

日本国特許庁

PATENT OFFICE
JAPANESE GOVERNMENT

31.03.99

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: 1998年 3月31日

REC'D 28 MAY 1999

出願番号
Application Number: 平成10年特許願第103721号

WIPO PCT

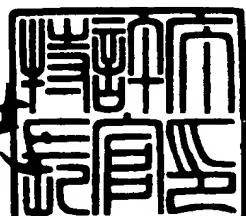
出願人
Applicant(s): セイコーエプソン株式会社

PRIORITY
DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

1999年 5月14日

特許庁長官
Commissioner,
Patent Office

伴佐山 建志



出証番号 出証特平11-3028510

【書類名】 特許願
【整理番号】 EP115301
【提出日】 平成10年 3月31日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 11/22
G06F 11/28

【発明の名称】 マイクロコンピュータ、電子機器及びデバッグシステム
【請求項の数】 12
【発明者】
【住所又は居所】 長野県諏訪市大和3丁目3番5号 セイコーホームズ株式会社内
【氏名】 工藤 真
【発明者】
【住所又は居所】 長野県諏訪市大和3丁目3番5号 セイコーホームズ株式会社内
【氏名】 土方 陽一
【特許出願人】
【識別番号】 000002369
【氏名又は名称】 セイコーホームズ株式会社
【代理人】
【識別番号】 100090479
【弁理士】
【氏名又は名称】 井上 一
【電話番号】 03-5397-0891
【代理人】
【識別番号】 100090387
【弁理士】
【氏名又は名称】 布施 行夫
【電話番号】 03-5397-0891

【代理人】

【識別番号】 100090398

【弁理士】

【氏名又は名称】 大渕 美千栄

【電話番号】 03-5397-0891

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9402500

【書類名】 明細書

【発明の名称】 マイクロコンピュータ、電子機器及びデバッグシステム

【特許請求の範囲】

【請求項1】 オンチップデバッグ機能を有するマイクロコンピュータであ
つて、

命令の実行処理を行う中央処理ユニットと、
マイクロコンピュータの外部に設けられデバッグコマンドを少なくとも1つの
プリミティブコマンドに変換するための処理を行う第2のモニタ手段との間でデ
ータを送受信し、実行するプリミティブコマンドを前記第2のモニタ手段からの
受信データに基づいて決定し、決定したプリミティブコマンドを実行するための
処理を行う第1のモニタ手段とを含むことを特徴とするマイクロコンピュータ。

【請求項2】 請求項1において、

前記プリミティブコマンドが、
ユーザプログラムの実行を開始するコマンド、デバッグモード時におけるメモ
リマップ上のアドレスにデータをライトするコマンド、及び前記メモリマップ上
のアドレスからデータをリードするコマンドを含むことを特徴とするマイクロコ
ンピュータ。

【請求項3】 請求項1又は2において、

前記中央処理ユニットの命令実行処理に使用され、デバッグモード時における
メモリマップ上にそのアドレスが割り付けられる制御レジスタを含むことを特徴
とするマイクロコンピュータ。

【請求項4】 請求項1乃至3のいずれかにおいて、

前記中央処理ユニットの内部レジスタの内容が退避され、デバッグモード時
におけるメモリマップ上にそのアドレスが割り付けられるモニタRAMを含むこと
を特徴とするマイクロコンピュータ。

【請求項5】 請求項1乃至4のいずれかにおいて、

前記第2のモニタ手段との間で半2重の双方向通信を行うための1本の双方向
通信ラインが接続される端子を含み、

スレーブとなる前記第1のモニタ手段が、
マスタとなる前記第2のモニタ手段からデータを受信したことを条件に、該受
信データに対応する処理を行い該受信データに対応する応答データを前記第2の
モニタ手段に送信することを特徴とするマイクロコンピュータ。

【請求項6】 請求項1乃至5のいずれかにおいて、

~~前記第2のモニタ手段からの受信データが、前記第1のモニタ手段が実行する~~
プリミティブコマンドの識別データを含むことを特徴とするマイクロコンピュー
タ。

【請求項7】 請求項1乃至6のいずれかにおいて、

前記第1のモニタ手段が、
前記第2のモニタ手段との間で固定長のデータを送受信することを特徴とする
マイクロコンピュータ。

【請求項8】 請求項1乃至7のいずれかにおいて、

前記第1のモニタ手段の処理を実行するためのモニタプログラムがROMに格
納されていることを特徴とするマイクロコンピュータ。

【請求項9】 請求項1乃至8のいずれかにおいて、

前記第1のモニタ手段が、
第1のクロックを分周し、調歩同期式で送受信されるデータの各ビットをサン
プリングするための第1のサンプリングクロックを生成する第1の分周回路と、
前記第1のサンプリングクロックに基づいてデータを送受信する回路とを含み

前記第1のモニタ手段が、
前記第2のモニタ手段が含む第2の分周回路が第2のサンプリングクロックを
生成するための信号として、前記第1のクロックを、前記第2のモニタ手段に供
給することを特徴とするマイクロコンピュータ。

【請求項10】 請求項1乃至9のいずれかにおいて、

前記第1のモニタ手段がリード及びライト可能なモニタRAMを含み、
前記第1のモニタ手段が、
ユーザプログラムの実行がブレークしデバッグモードに移行した場合に、前記

中央処理ユニットのプログラムカウンタ値及び内部レジスタの内容を前記モニタRAMに待避することを特徴とするマイクロコンピュータ。

【請求項11】 請求項1乃至10のいずれかのマイクロコンピュータと、前記マイクロコンピュータの処理対象となるデータの入力源と、前記マイクロコンピュータにより処理されたデータを出力するための出力装置とを含むことを特徴とする電子機器。

【請求項12】 マイクロコンピュータを含むターゲットシステムのためのデバッグシステムであって、

ホストシステムが発行したデバッグコマンドを少なくとも1つのプリミティブコマンドに変換するための処理を行う第2のモニタ手段と、

前記第2のモニタ手段との間でデータを送受信し、実行するプリミティブコマンドを前記第2のモニタ手段からの受信データに基づいて決定し、決定したプリミティブコマンドを実行するための処理を行う第1のモニタ手段とを含むことを特徴とするデバッグシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、マイクロコンピュータ、マイクロコンピュータを含む電子機器、及びデバッグシステムに関する。

【0002】

【背景技術及び発明が解決しようとする課題】

近年、ゲーム装置、カーナビゲーションシステム、プリンタ、携帯情報端末などの電子機器に組み込まれ、高度な情報処理を実現できるマイクロコンピュータに対する需要が高まっている。そして、このような組み込み型のマイクロコンピュータは、通常、ターゲットシステムと呼ばれるユーザボードに実装される。そして、このターゲットシステムを動作させるソフトウェアの開発を支援するためにはICE (In-Circuit Emulator) と呼ばれるソフトウェア開発支援ツールが広く使用されている。

【0003】

さて、このようなICEとしては、従来、図1 (A) に示すようなCPU置き換え型と呼ばれるICEが主流を占めていた。このCPU置き換え型ICEでは、デバッグ時にターゲットシステム300からマイクロコンピュータ302を取り外し、その代わりにデバッグツール304のプローブ306を接続する。そして、このデバッグツール304に、取り外したマイクロコンピュータ302の動作をエミュレートさせる。また、このデバッグツール304に、デバッグのため必要な種々の処理を行わせる。

【0004】

しかしながら、このCPU置き換え型ICEには、プローブ306のピン数が多くなると共にプローブ306の線308が増えるという欠点があった。このため、マイクロコンピュータ302の高周波数動作をエミュレートすることが困難になる（例えば33MHz程度が限界）。またターゲットシステム300の設計も困難になる。更に、マイクロコンピュータ302を実装して動作させる実動作時とデバッグツール304でマイクロコンピュータ302の動作をエミュレートするデバッグモード時とで、ターゲットシステム300の動作環境（信号のタイミング、負荷条件）が変化してしまう。またこのCPU置き換え型ICEには、マイクロコンピュータが異なれば、たとえそれが派生品であっても、設計が異なるデバッグツールや、ピン数やピンの位置が異なるプローブを使用しなければならないという問題もあった。

【0005】

一方、このようなCPU置き換え型ICEの欠点を解消するものとして、図1 (B) に示すような、モニタプログラム310をターゲットシステム312に実装するタイプのICEが知られている。しかしながら、これまでのモニタプログラム実装型ICEでは、モニタプログラム310に、全てのデバッグコマンド（プログラムのロード、GO、ステップ実行、メモリのリード・ライト、内部レジスタのリード・ライト、ブレークポイントの設定・解除）を実行する機能を持たせる必要があった。したがって、モニタプログラム310の命令コードサイズが非常に大きくなってしまう（例えば30～50Kバイト）。このため、ユーザが自由に使用できるメモリ領域が減ってしまうと共に、デバッグ時とデバッグ時以

外でシステムが異なってしまうという問題があった。図1 (B) の問題点を解決する方法として、モニタプログラムをチップ上にのせるオンチップデバッグという方法があるが、命令コードサイズが大きなモニタプログラムをチップ内に設けると、チップのサイズを大きくしてしまうという問題点がある。

【0006】

本発明は、以上のような技術的課題に鑑みてなされたものであり、その目的とするところは、小さな命令コードサイズ又は回路規模でオンチップデバッグ機能を実現できるマイクロコンピュータ、これを含む電子機器、及びデバッグシステムを提供することにある。

【0007】

【課題を解決するための手段】

上記課題を解決するために本発明は、オンチップデバッグ機能を有するマイクロコンピュータであって、命令の実行処理を行う中央処理ユニットと、マイクロコンピュータの外部に設けられデバッグコマンドを少なくとも1つのプリミティブコマンドに変換するための処理を行う第2のモニタ手段との間でデータを送受信し、実行するプリミティブコマンドを前記第2のモニタ手段からの受信データに基づいて決定し、決定したプリミティブコマンドを実行するための処理を行う第1のモニタ手段とを含むことを特徴とする。

【0008】

本発明によれば、マイクロコンピュータの外部に設けられた第2のモニタ手段が、ホストシステム等が発行したデバッグコマンドをプリミティブコマンドに変換(分解)するための処理を行う。そして、第1のモニタ手段は、この第2のモニタ手段からデータを受信し、この受信データに基づいて決定されたプリミティブコマンドを実行するための処理を行う。本発明によれば、第1のモニタ手段の処理を実行するためのモニタプログラムに、各デバッグコマンドを実行するための複雑なルーチンを持たせる必要がなくなる。したがって、モニタプログラムの命令コードサイズを格段に小さくできるようになり、小さなハードウェア規模でオンチップデバッグ機能を実現できるようになる。

【0009】

また本発明は、前記プリミティブコマンドが、ユーザプログラムの実行を開始するコマンド、デバッグモード時におけるメモリマップ上のアドレスにデータをライトするコマンド、及び前記メモリマップ上のアドレスからデータをリードするコマンドを含むことを特徴とする。このようにプリミティブコマンドをシンプルにすることで、モニタプログラムの命令コードサイズを更に小さくできるようになる。

【0010】

また本発明は、前記中央処理ユニットの命令実行処理に使用され、デバッグモード時におけるメモリマップ上にそのアドレスが割り付けられる制御レジスタを含むことを特徴とする。このようにすれば、デバッグモード時に制御レジスタを用いたデバッグ処理を行うことが可能になり、処理の簡易化、ハードウェアの小規模化を図れるようになる。

【0011】

また本発明は、前記中央処理ユニットの内部レジスタの内容が退避され、デバッグモード時におけるメモリマップ上にそのアドレスが割り付けられるモニタRAMを含むことを特徴とする。このようにすれば、デバッグモード時に、内部レジスタの内容をリードしたりすることが可能になり、より多様なデバッグ機能を実現できるようになる。

【0012】

また本発明は、前記第2のモニタ手段との間で半2重の双方向通信を行うための1本の双方向通信ラインが接続される端子を含み、スレーブとなる前記第1のモニタ手段が、マスタとなる前記第2のモニタ手段からデータを受信したことを条件に、該受信データに対応する処理を行い該受信データに対応する応答データを前記第2のモニタ手段に送信することを特徴とする。このようにすれば、マイクロコンピュータの端子（ピン）数を減らすことが可能となり、マイクロコンピュータの低コスト化を図れるようになる。また、第1、第2のモニタ手段の間の通信プロトコルを単純化でき、モニタプログラムの命令コードサイズを更に小さくできるようになる。

【0013】

また本発明は、前記第2のモニタ手段からの受信データが、前記第1のモニタ手段が実行するプリミティブコマンドの識別データを含むことを特徴とする。このようにすることで、プリミティブコマンドの実行の指示を簡易に第2のモニタ手段から第1のモニタ手段に伝えることが可能になる。

【0014】

また本発明は、前記第1のモニタ手段が、前記第2のモニタ手段との間で固定長のデータを送受信することを特徴とする。このようにすることで、第1のモニタ手段のモニタプログラムの命令コードサイズを更に小さくすることが可能になる。

【0015】

また本発明は、前記第1のモニタ手段の処理を実行するためのモニタプログラムがROMに格納されていることを特徴とする。このようにすることで、RAMに比べて占有面積の小さいROMにモニタプログラムが格納されると共に、モニタプログラムをRAMにロードするロジック回路等が不要になる。したがって、マイクロコンピュータの更なる小規模化を図れるようになる。

【0016】

また本発明は、前記第1のモニタ手段が、第1のクロックを分周し、調歩同期式で送受信されるデータの各ビットをサンプリングするための第1のサンプリングクロックを生成する第1の分周回路と、前記第1のサンプリングクロックに基づいてデータを送受信する回路とを含み、前記第1のモニタ手段が、前記第2のモニタ手段が含む第2の分周回路が第2のサンプリングクロックを生成するための信号として、前記第1のクロックを、前記第2のモニタ手段に供給することを特徴とする。このように、サンプリングクロックを生成するための第1のクロックを第1、第2のモニタ手段で共有することで、通信データのサンプリングエラーの発生率を格段に低くすることが可能になると共に、通信速度の最適化、高速化を図れるようになる。

【0017】

また本発明は、前記第1のモニタ手段がリード及びライト可能なモニタRAMを含み、前記第1のモニタ手段が、ユーザプログラムの実行がブレークしデバッ

グモードに移行した場合に、前記中央処理ユニットのプログラムカウンタ値及び内部レジスタの内容を前記モニタRAMに待避することを特徴とする。このようにすることで、デバッグモードからユーザプログラム実行モードに戻った時に適正にユーザプログラムを実行できるようになる。また第1のモニタ手段が内部レジスタの内容等を利用して各種の処理を行うことも可能になる。

【0018】

また本発明に係る電子機器は、上記のいずれかのマイクロコンピュータと、前記マイクロコンピュータの処理対象となるデータの入力源と、前記マイクロコンピュータにより処理されたデータを出力するための出力装置とを含むことを特徴とする。このようにすれば、電子機器を動作させるプログラムなどのデバッグ作業の効率化を図れるようになり、電子機器の開発期間の短縮化、低コスト化を図れるようになる。

【0019】

また本発明は、マイクロコンピュータを含むターゲットシステムのためのデバッグシステムであって、ホストシステムが発行したデバッグコマンドを少なくとも1つのプリミティブコマンドに変換するための処理を行う第2のモニタ手段と、前記第2のモニタ手段との間でデータを送受信し、実行するプリミティブコマンドを前記第2のモニタ手段からの受信データに基づいて決定し、決定したプリミティブコマンドを実行するための処理を行う第1のモニタ手段とを含むことを特徴とする。

【0020】

本発明によれば、第1のモニタ手段の処理を実行するためのモニタプログラムの命令コードサイズを格段に小さくできる。これにより、ユーザが自由に使用できるメモリ領域を増やすことが可能になる。また、ターゲットシステムを、実動作時の環境と同一の環境でデバッグできるデバッグシステムを提供できるようになる。

【0021】

【発明の実施の形態】

以下、本発明の好適な実施形態について図面を用いて詳細に説明する。

【0022】

1. 本実施形態の特徴

まず本実施形態の特徴について図2を用いて説明する。

【0023】

図2に示すように、本実施形態では、マイクロコンピュータ10が、CPU(中央処理ユニット)12及び本実施形態の要部であるミニモニタ部(第1のモニタ手段)14を含む。また、マイクロコンピュータ10の外部にはメインモニタ部(第2のモニタ手段)16が設けられている。ここでメインモニタ部16は、例えばホストシステムなどが発行したデバッグコマンドをプリミティブコマンドに変換(分解)するための処理を行う。また、ミニモニタ部14は、メインモニタ部16との間でデータを送受信する。そして、ミニモニタ部14は、実行するプリミティブコマンドを、メインモニタ部16からの受信データに基づいて決定し、プリミティブコマンドを実行するための処理を行う。

【0024】

ここで、メインモニタ部16の変換処理の対象となるデバッグコマンドとしては、プログラムロード、GO、ステップ実行、メモリライト、メモリリード、内部レジスタライト、内部レジスタリード、ブレークポイント設定、ブレークポイント解除などのコマンドを考えることができる。メインモニタ部16は、これらの多様で複雑なデバッグコマンドを、例えばGO、ライト(デバッグモード時ににおけるメモリマップ上の所与のアドレスへのライト)、リード(メモリマップ上の所与のアドレスからのリード)などの、シンプルでプリミティブなコマンドに変換する処理を行う。このようにすることで、ミニモニタ部14の処理を行うミニモニタプログラムの命令コードサイズを格段に小さくすることができる。これにより、マイクロコンピュータ10のオンチップデバッグ機能を実現できるようになる。

【0025】

即ち、図1(B)に示すようなタイプのICEでは、モニタプログラム310は、プログラムロード、GO、ステップ実行などのデバッグコマンドの全ての処理ルーチンを有している。このため、モニタプログラム310の命令コードサイ

ズが非常に大きくなり（例えば30～50Kバイト）、モニタプログラム310をマイクロコンピュータ314に内蔵することは事実上困難となる。

【0026】

これに対して、本実施形態では、ミニモニタ部14の処理を行うミニモニタプログラムは、GO、ライト、リードなどのシンプルなプリミティブコマンドの処理ルーチンのみを有し、命令コードサイズが非常に小さい（例えば256バイト）。したがって、ミニモニタプログラムをマイクロコンピュータ10に内蔵することが可能となり、オンチップデバッグ機能を実現できるようになる。また、ユーザが自由に使用できるメモリ領域の減少を最小限に抑える、あるいは全く減少させないようにすることが可能になる。

【0027】

2. 詳細な構成例

図3に本実施形態のマイクロコンピュータ及びデバッグシステムの詳細な構成例を示す。図3に示すように、マイクロコンピュータ20は、CPU22、BCU（バス制御ユニット）26、内部メモリ（ミニモニタROM42及びミニモニタRAM44以外の内部ROM及び内部RAM）28、クロック生成部30、ミニモニタ部40（第1のモニタ手段）、トレース部50を含む。

【0028】

ここでCPU22は、種々の命令の実行処理を行うものであり、内部レジスタ24を含む。内部レジスタ24は、汎用レジスタであるR0～R15や、特殊レジスタであるSP（スタックポインタレジスタ）、AHR（積和結果データの上位レジスタ）、ALR（積和結果データの下位レジスタ）などを含む。

【0029】

BCU26はバスを制御するものである。例えば、CPU22に接続されるハーバードアーキテクチャのバス31や、内部メモリ28に接続されるバス32や、外部メモリ36に接続される外部バス33や、ミニモニタ部40、トレース部50などに接続される内部バス34の制御を行う。

【0030】

またクロック生成部30は、マイクロコンピュータ20内で使用される各種の

クロックを生成するものである。クロック生成部30はBCLKを介して外部のデバッグツール60にもクロックを供給している。

【0031】

ミニモニタ部40は、ミニモニタROM42、ミニモニタRAM44、制御レジスタ46、SIO48を含む。

【0032】

ここで、ミニモニタROM42には、ミニモニタプログラムが格納される。本実施形態では、このミニモニタプログラムは、GO、リード、ライトなどのシンプルでプリミティブなコマンドの処理のみを行うようになっている。このため、ミニモニタROM42のメモリ容量を例えば256バイト程度に抑えることができ、オンチップデバッグ機能を持たせながらマイクロコンピュータ20を小規模化できるようになる。

【0033】

ミニモニタRAM44には、デバッグモードへの移行時に（ユーザプログラムのブレーク発生時に）、CPU22の内部レジスタ24の内容が退避される。これにより、デバッグモードの終了後にユーザプログラムの実行を適正に再スタートできるようになる。また内部レジスタの内容のリード等を、ミニモニタプログラムが持つプリミティブなリードコマンド等で実現できるようになる。

【0034】

制御レジスタ46は、各種のデバッグ処理を制御するためのレジスタであり、ステップ実行イネーブルビット、ブレークイネーブルビット、ブレークアドレスビット、トレースイネーブルビットなどを有する。ミニモニタプログラムにより動作するCPU22が制御レジスタ46の各ビットにデータをライトしたり、各ビットのデータをリードすることで、各種のデバッグ処理が実現される。

【0035】

SIO48は、マイクロコンピュータ20の外部に設けられたデバッグツール60との間でデータを送受信するためのものである。SIO48とデバッグツール60との間は、TXD/RXD（データ送受信ライン）で接続されている。

【0036】

トレース部50は、リアルタイムトレース機能を実現するためのものである。トレース部50とデバッグツール60との間は、CPU22の命令実行のステートを表す3ビットのDST[2:0]と、分岐先のPC(プログラムカウンタ)値を表すDPCOという4本のラインで接続されている。

【0037】

デバッグツール60はメインモニタ部62を含み、パーソナルコンピュータ等により実現されるホストシステム66に接続される。ホストシステム66が、ユーザの操作により、プログラムロード、ステップ実行などのデバッグコマンドを発行すると、メインモニタ部62が、このデバッグコマンドをプリミティブコマンドに変換(分解)するための処理を行う。そして、メインモニタ部62が、プリミティブコマンドの実行を指示するデータをミニモニタ部40に送信すると、ミニモニタ部40が、指示されたプリミティブコマンドを実行するための処理を行うことになる。

【0038】

図4に、デバッグモード時のメモリマップの例を示す。図4のD1、D2、D3に示すように、デバッグモード時には、図3の制御レジスタ46、ミニモニタRAM44、ミニモニタROM42のアドレスも、メモリマップ上に割り付けられる。

【0039】

3. プリミティブコマンドへの変換

図5(A)、(B)、(C)、(D)に、各種のデバッグコマンドをプリミティブコマンドへ変換する処理について模式的に示す。

【0040】

例えば図5(A)に示すように、(ADD…、SUB…、AND…、OR…、XOR…、LD. W….)という12バイトのプログラムを80010h番地にロードするというデバッグコマンドが発行されたとする。この場合、このプログラムロードコマンドは、ライト(80010h、ADD…、SUB)、ライト(80014h、AND…、OR….)、ライト(80018h、XOR…、LD. W….)という3つのプリミティブなライトコマンドに変換される。即

ち、ミニモニタプログラムが、この3つのプリミティブなライトコマンドを実行することで、プログラムロードコマンドが実現されるようになる。

【0041】

また図5 (B) に示すようにステップ実行コマンドというデバッグコマンドが発行されたとする。すると、このステップ実行コマンドは、図3の制御レジスタ46のステップ実行イネーブルビットへのライトコマンド(図4のD1のアドレスへのライトコマンド)とGOコマンドに変換される。即ち、ミニモニタプログラムが、このプリミティブなライトコマンドとGOコマンドを実行することで、ステップ実行コマンドが実現されるようになる。

【0042】

また図5 (C) に示すように内部レジスタリードコマンドというデバッグコマンドが発行されたとする。すると、この内部レジスタリードコマンドは、メモリマップ上のミニモニタRAM44(内部レジスタの内容の退避先)からのリードコマンド(図4のD2のアドレスからのリードコマンド)に変換される。即ち、ミニモニタプログラムが、このプリミティブなリードコマンドを実行することで、内部レジスタリードコマンドが実現されるようになる。内部レジスタライトコマンド、メモリリードコマンド、メモリライトコマンドも同様にして実現される。

【0043】

また図5 (D) に示すようにブレークポイント設定コマンドというデバッグコマンドが発行されたとする。すると、このブレークポイント設定コマンドは、制御レジスタ46のブレークイネーブルビット及びブレークアドレスビットへのライトコマンドに変換される。即ち、ミニモニタプログラムが、このプリミティブなライトコマンドを実行することで、ブレークポイント設定コマンドが実現されるようになる。

【0044】

このように本実施形態では、複雑で多様なデバッグコマンドが、プリミティブでシンプルなリード、ライト、GOコマンドに変換される。そして、ミニモニタプログラムは、このプリミティブなリード、ライト、GOコマンドを実行するだ

けでよいため、ミニモニタプログラムの命令コードサイズは非常に小さくなる。この結果、ミニモニタROM42のメモリ容量も小さくでき、小さなハードウェア規模でオンチップデバッグ機能を実現できるようになる。

【0045】

4. S I Oの構成例

図6にS I O48の構成例を示す。S I O48は、送受信バッファ70、シフトレジスタ76、送受信切替部78、クロック制御部80及び制御レジスタ84を含む。

【0046】

ここで送受信バッファ70は、送信データ、受信データを一時的に蓄えるためのものであり、送信バッファ72、受信バッファ74を有する。シフトレジスタ76は、送信バッファ72からの送信データをパラレルデータからシリアルデータに変換し送受信切替部78に出力する機能を有する。また送受信切替部78からの受信データをシリアルデータからパラレルデータに変換し受信バッファ74に出力する機能も有する。送受信切替部78は、データの送信と受信とを切り替えるためのものである。これにより、T X D / R X Dを使用した半2重のデータ送受信が可能になる。

【0047】

クロック制御部80は、内蔵する分周回路82によりB C L Kを分周し、この分周により得られたサンプリングクロックSMC1をシフトレジスタ76に出力する。シフトレジスタ76は、このSMC1に基づき動作する。またクロック制御部80は、デバッグツール60にB C L Kを供給する。これにより、マイクロコンピュータ20とデバッグツール60により、B C L Kが共有されるようになる。

【0048】

分周回路82での分周比は制御レジスタ84により設定される。即ちC P U22により実行されるミニモニタプログラムが、所望の分周比を制御レジスタ84に書き込むことで、分周回路82での分周比が設定されることになる。なお、制御レジスタ84のアドレスも、図3の制御レジスタ46と同様に、図4のD1の

位置に割り付けられている。

【0049】

5. デバッグツールの構成例

図7にデバッグツール60の構成例を示す。

【0050】

CPU90は、ROM108に格納されるプログラムを実行したり、デバッグツール60の全体の制御を行うものである。送受信切替部92は、データの送信と受信とを切り替えるためのものである。クロック制御部94は、CPU90のSCLK端子、アドレスアップカウンタ100、トレースメモリ104に供給するクロックを制御するものである。このクロック制御部94には、マイクロコンピュータ20(SIO48)からのBCLKが入力される。クロック制御部94は周波数検出回路95、分周回路96を含む。周波数検出回路95は、BCLKの周波数が属する周波数範囲を検出して、その結果を制御レジスタ98に出力する。また分周回路96での分周比は制御レジスタ98により制御される。即ちCPU90により実行されるメインモニタプログラム(メインモニタROM110に格納)が、制御レジスタ98からBCLKの周波数範囲を読み出す。そして、メインモニタプログラムは、この周波数範囲に応じた最適な分周比を決定し、この分周比を制御レジスタ98に書き込む。そして、分周回路96は、この分周比でBCLKを分周してSMC2を生成し、CPU90のSCLK端子に出力する。

【0051】

アドレスアップカウンタ100は、トレースメモリのアドレスをカウントアップするためのものである。セレクタ102は、ライン122(アドレスアップカウンタ100が出力するアドレス)とライン124(アドレスバス120からのアドレス)のいずれかを選択し、トレースメモリ104のアドレス端子にデータを出力する。またセレクタ106は、ライン126(図3のトレース部50の出力であるDST[2:0]、DPCO)とライン128(データバス118)のいずれかを選択し、トレースメモリ104のデータ端子にデータを出力したり、データ端子からデータを取り出す。

【0052】

ROM108はメインモニタROM110（図3のメインモニタ部62に相当）を含み、メインモニタROM110には、メインモニタプログラムが格納される。このメインモニタプログラムは、図5（A）～図5（D）で説明したように、デバッグコマンドをプリミティブコマンドに変換するための処理を行う。RAM112は、CPU90のワーク領域となるものである。

【0053】

RS232Cインターフェース114、パラレルインターフェース116は、図3のホストシステム66とのインターフェースとなるものであり、ホストシステム66からのデバッグコマンドはこれらのインターフェースを介してCPU90に入力されることになる。クロック生成部118は、CPU90を動作させるクロックなどを生成するものである。

【0054】

次に本実施形態でのリアルタイムトレース処理について簡単に説明する。本実施形態では、図3のCPU22の命令実行のステートを表す3ビットのDST[2:0]と、分岐先のPC値を表すDPCOをトレースメモリ104に蓄える。そして、トレースメモリ104に蓄えられたデータと、ユーザプログラムのソースコードとに基づいて、トレースデータを合成する。このようにすることで、マイクロコンピュータ20とデバッグツール60との間の接続ラインの本数を少なくしながら、リアルタイムトレース機能を実現することが可能になる。

【0055】

ユーザプログラム実行モードにおいては、ライン122が選択され、セレクタ102を介してアドレスアップカウンタ100の出力がトレースメモリ104のアドレス端子に入力される。また、ライン126が選択され、セレクタ106を介してDST[2:0]、DPCOがトレースメモリ105のデータ端子に入力される。ここでアドレスアップカウンタ100には、まず最初に、データバス118、アドレスバス120を用いてCPU90により図8（A）に示すようなスタートアドレスが設定される。またアドレスアップカウンタ100のST/SP（開始/停止）端子には、トレース範囲を特定するDST[2]のラインが接続

される。そして図8 (B) に示すように、DST [2] のラインに第1のパルス130が入力されると、アドレスアップカウンタ100のアドレスアップカウントが開始する。そして、DST [2] のラインに第2のパルス132が入力されると、アドレスアップカウンタ100のアドレスアップカウントが停止し、トレース動作が停止する。このようにして、所望のトレース範囲でのデータ (DST [2:0]、DPCO) をトレースメモリ104に蓄えることが可能になる。

【0056】

一方、ユーザプログラム実行モードからデバッグモードに移行すると、ライン124が選択され、セレクタ102を介してアドレスバス120からのアドレスがトレースメモリ104のアドレス端子に入力される。またライン128が選択され、セレクタ106を介してトレースメモリ104からのデータがデータバス118に出力される。これにより、トレースメモリ104に蓄えられたデータ (DST [2:0]、DPCO) を、デバッグモード時にCPU90 (メインモニタプログラム) が読み出すことが可能になる。そして、読み出されたデータとユーザープログラムのソースコードとに基づいて、トレースデータを合成することが可能になる。

【0057】

6. データの送受信

さて、図9 (A) に示すように、ミニモニタ部40とメインモニタ部62の間でのデバッグデータの通信の手法としては、TXD (送信) とRXD (受信) のラインを別々に設け、全2重で通信する手法が考えられる。

【0058】

しかしながら、このようにデバッグデータの通信のために2本のライン (端子) を使用してしまうと、その分だけマイクロコンピュータの端子数 (ピン数) が増えてしまい、これはマイクロコンピュータの高コスト化を招く。

【0059】

そこで本実施形態では、図9 (B) に示すように、ミニモニタ部40とメインモニタ部62との間に1本のTXD/RXDライン (双方向通信ライン) を設け、半2重の双方向通信を行う。このようにすることで、マイクロコンピュータの

端子数の増加を最小限に抑えることができ、マイクロコンピュータの低コスト化を図れる。

【0060】

そして更に本実施形態では、図9 (C) に示すように、スレーブとなるミニモニタ部40が、マスタとなるメインモニタ部62からデータを受信したことを条件に、その受信データに対応する処理を行い、その受信データに対応する応答データをメインモニタ部62に送信する。即ち、メインモニタ部62がデータ（コマンド）をミニモニタ部40に送信すると、ウェイト状態となっていたミニモニタ部40がこれを受信し、その受信データに対応する処理を行う。そして、その受信データに対応するデータ（リプライ）をメインモニタ部62に送信する。その後、ミニモニタ部40は、メインモニタ部62からデータを受信するまでウェイト状態になる。即ちミニモニタ部40は、メインモニタ部62からデータを受信するまで動作を停止し、データを受信したことを条件に動作を開始する。このようにすることで、ミニモニタ部40とメインモニタ部62との間の通信ラインを1本にしながら、データを適正に送受信できるようになる。

【0061】

さて、図9 (A) の通信手法は、TxDとRxDが別ラインになっているため、図9 (B) に比べてデータを高速に通信できるという利点がある。また、ミニモニタ部40、メインモニタ部62の一方で通信エラーが発生した場合に、即座にエラーメッセージを他方に返すことができるという利点もある。例えば、ミニモニタ部40で通信エラーが発生した場合には、RxDでのデータ受信の終了を待つことなくTxDを使用して即座にエラーメッセージをメインモニタ部62に返すことができる。

【0062】

一方、本実施形態では後述するように、ミニモニタ部40とメインモニタ部62とでBCLKを共有させている。これにより後述するように、最適な速度で高速にデータを通信できるようになる。したがって、図9 (A) のような通信ラインを2本設ける構成にせずに、図9 (B) のように通信ラインを1本だけ設ける構成にしても、問題なくデバッグデータを高速に送受信できるようになる。

【0063】

また、本実施形態では後述するように、送受信データは固定長でありデータ長が短い（例えば14バイト）。したがって、例えばミニモニタ部40で通信エラーが発生した場合、受信処理の終了を待ってからエラーメッセージを送信しても、時間的な遅れはそれほど大きくならない。また、送受信データのデータ長が短いため、通信エラーの発生自体も最小限に抑えることができる。

【0064】

このように本実施形態では、デバッグデータの通信ラインを1本だけ設ける構成にすることでマイクロコンピュータの端子数を減らすという利益を得ている。そしてこの利益を得るために生じる不利益（通信速度の低下、エラーメッセージの遅延）を、BCLKを共有化したり送受信データを固定長の短いデータにすることで解消している。

【0065】

7. 送受信データのフォーマット及び種類

図10 (A) に、TXD/RXDのラインで送受信されるデータのフォーマット例を示す。送受信データは、1バイトのID（コマンド識別データ）、1バイトのデータサイズ、4バイトのアドレス、4バイトのデータ1、4バイトのデータ2から成る14バイトの固定長データになっている。

【0066】

図10 (B) に示すように、メインモニタ部62がミニモニタ部40にGOコマンドの実行を指示する場合、ミニモニタ部40が受信するデータのIDには、GOコマンドの識別データである00hが設定される。そしてこの場合には、ミニモニタ部40は、メインモニタ部62にデータを送信しない。

【0067】

図10 (C) に示すように、メインモニタ部62がライトコマンドの実行を指示する場合、ミニモニタ部40の受信データのIDには、ライトコマンドの識別データである01hが設定される。また受信データのアドレス、データ1、データ2には、各々、ライトアドレス、ライトデータ1、ライトデータ2が設定される。そして、ミニモニタ部40の送信データのデータ1には01hが設定される

【0068】

なお、ライトコマンドを、バイトデータのライトコマンド、ハーフワードデータのライトコマンド、ワードデータのライトコマンド、ダブルワードデータのライトコマンドというように、ライトするデータの長さに応じて複数種類用意するようにしてもよい。このようにした場合には、各々のライトコマンドに対して異なるIDを割り振ることになる。

【0069】

図10 (D) に示すように、メインモニタ部62がリードコマンドの実行を指示する場合、ミニモニタ部40の受信データのIDには、リードコマンドの識別データである02hが設定される。また受信データのアドレスにはリードアドレスが設定される。そして、ミニモニタ部40の送信データのデータ1、データ2には、各々、リードコマンド処理で得られたリードデータ1、リードデータ2が設定される。

【0070】

さて、本実施形態では、ミニモニタ部62が実行するプリミティブコマンドとして、GO、ライト、リードコマンド以外にも、外部ルーチンジャンプ、データフィルなどのコマンドを用意している。

【0071】

ここで外部ルーチンジャンプコマンドは、外部ルーチンにジャンプすることを指示するコマンドである。このような外部ルーチンジャンプコマンドを用いることで、例えばフラッシュメモリ(EEPROM)の初期化プログラムや書き込みプログラムのルーチンへのジャンプが可能になる。そして図11 (A) に示すように、メインモニタ部62が外部ルーチンジャンプコマンドの実行を指示する場合、ミニモニタ部40の受信データのIDには、外部ルーチンジャンプコマンドの識別データである03hが設定される。また受信データのデータサイズ、アドレス、データ1、データ2には、各々、18h(誤動作防止チェック)、%R12(ルーチンアドレス)、%R13(ライトデータ)、%R14(データアドレス)が設定される。そして、ミニモニタ部40の送信データのデータ1には、%

R10 (戻り値。戻り値が0の場合に正常終了) が設定される。

【0072】

またデータファイルコマンドは、メモリを所与の値（例えば0）でファイルするためのコマンドである。例えば大容量のメモリの全てのビットを0の値に設定する場合、ライトコマンドを用いたのでは処理時間が非常に長くなる。このような場合、データファイルコマンドが有効になる。そして図1-1 (B) に示すように、メインモニタ部62がデータファイルコマンドの実行を指示する場合、ミニモニタ部40の受信データのIDには、データファイルコマンドの識別データである04hが設定される。また受信データのデータサイズ、アドレス、データ1、データ2には、各々、データサイズである1、2又は4、スタートアドレス、ファイル回数、ファイルパターンが設定される。

【0073】

このように本実施形態では、メインモニタ部62からの受信データに、ミニモニタ部40が実行するプリミティブコマンドの識別データIDを含ませている。このようにすることで、プリミティブコマンドの実行の指示を簡易にミニモニタ部40に伝えることが可能になる。

【0074】

8. 固定長の送受信データ

本実施形態では図10 (A) に示すように、ミニモニタ部40、メインモニタ部62間の送受信データを14バイトの固定長のデータにしている。このようにすることで、ミニモニタプログラムの命令コードサイズを更に小さくすることが可能になる。

【0075】

即ち送受信データを可変長にすると、図12 (A) のE1、E2、E3に示す処理部分（命令列）が、ほとんど全てのコマンドについて必要になる。これらのE1、E2、E3の処理部分は、どれだけのデータ数の処理が必要かを判断する部分である。即ち可変長データを処理する場合には、送受信データに含まれるデータサイズに基づいて、処理すべきデータの数を調べる必要がある。そして、そのデータ数を、例えばワーク領域であるRAM上に保持しておいて、1つのデー

タの処理が終了する毎にこのデータ数をデクリメントしたり、データ数が0になったか否かを判断したりする処理が必要になる。このため、図12(A)に示すように、ミニモニタプログラムのソースコードのサイズが大きくなる。

【0076】

これに対して、本実施形態では、送受信データを固定長にしている。このため、図12(A)と図12(B)を比較すればわかるように、ミニモニタプログラムのソースコードのサイズを可変長の場合に比べて例えば2/3程度にできる。これにより、図3のミニモニタROM42のサイズを更に小さくでき、小さなハードウェア規模でオンチップデバッグ機能を実現できるようになる。

【0077】

なお、送受信データを固定長にすると通信効率が悪化するため、可変長にする場合に比べて通信速度が低下するという問題が生じる。そこで、本実施形態では後述するように、ミニモニタ部40とメインモニタ部62にBCLKを共有させている。これにより、最適な速度で高速にデータを通信できるようになり、上記の問題を解消できる。

【0078】

9. ミニモニタROM

本実施形態では図3に示すように、マイクロコンピュータ20内にミニモニタROM42を設け、このミニモニタROM42にミニモニタプログラムを格納するようにしている。このようにすることで、ハードウェアの小規模化、マイクロコンピュータの低コスト化を図れるようになる。

【0079】

例えば図13(A)に示す構成では、マイクロコンピュータ330にローダロジック回路332及びRAM334を設ける。そして、このローダロジック回路332を用いて、JTAGインターフェース336を介して外部からRAM334にモニタプログラムをロードする。しかしながら、この構成では、ローダロジック回路332や、サイズがROMの5~10倍程度になるRAM334をマイクロコンピュータ330内に設ける必要がある。このため、マイクロコンピュータ330の大規模化、高コスト化の問題を招く。

【0080】

これに対して、本実施形態では図13（B）に示すように、ミニモニタプログラムは、サイズがRAMの1/5～1/10倍程度になるミニモニタROM42に格納される。またローダロジック回路も必要ない。したがって、図13（A）の構成に比べて、マイクロコンピュータ20の小規模化、低コスト化を図れるようになる。

【0081】

また図13（A）の構成では、電源投入時又はリセット時に、CPUを一旦停止し、次にローダロジック回路332でモニタプログラムをRAM334にロードし、その後、CPUをデバッグモードで再スタートする必要がある。このため、処理が複雑化すると共に、デバッグモードのスタートに時間を要するようになってしまう。

【0082】

これに対して、図13（B）の本実施形態では、RAMにミニモニタプログラムをロードする必要がない。このため、電源投入時又はリセット時に、CPUを一旦停止することなく、CPUのデバッグモードの動作をすぐにスタートできるようになる。

【0083】

10. BCLKの共有化

さて、マイクロコンピュータとデバッグツールとの間でのデータの通信方式としては、いわゆる同期式と呼ばれるものや調歩同期式と呼ばれるものを採用できる。そしてICEにおいては、マイクロコンピュータとデバッグツールとの間の通信ラインの本数をできる限り少なくすることが望まれる。更に通信データのサンプリングエラーをできる限り防止することも望まれる。

【0084】

ところが同期式で通信を行う場合には、図14（A）に示すように、マイクロコンピュータ340（第1の情報処理装置）とデバッグツール342（第2の情報処理装置）との間に4本の通信ラインを設ける必要がある。即ち、送信データであるTXDのライン、TXDのサンプリングクロックであるTCLKのライン

、受信データであるRXDのライン、RXDのサンプリングクロックであるCLKのラインが必要になる。このため、通信ラインの本数が不要に増えてしまう。

【0085】

一方、調歩同期式で通信を行う場合には、図14（B）に示すように、マイクロコンピュータ340とデバッグツール342は、別々にほぼ同一周波数のクロックを持つことになる。例えばマイクロコンピュータ340はクロックCLK1を持ち、デバッグシステム342はクロックCLK2を持ち、CLK1とCLK2の周波数をほぼ同一にする。そしてマイクロコンピュータ340は、図15（A）に示すように、CLK1を分周することでサンプリングクロックSMC1を生成し、調歩同期式で通信されるデータの各ビット（スタートビット、D0～D7、トップビット）を、このSMC1でサンプリングする。またデバッグツール342は、図15（B）に示すように、CLK2を分周することでサンプリングクロックSMC2を生成し、調歩同期式で通信されるデータの各ビット（スタートビット、D0～D7、トップビット）をこのSMC2でサンプリングする。

【0086】

しかしながら、この調歩同期式では、マイクロコンピュータ340に含まれるCPUの動作周波数が高くなりCLK1及びCLK2の周波数が高くなると、SMC1及びSMC2の周波数も高くなり、通信データのサンプリングエラーが生じやすくなってしまう。逆に言えば、通信データのサンプリングエラーが生じない程度までしか、CLK1及びCLK2の周波数を高くできない。これは、マイクロコンピュータ340が高速動作する環境でのデバッグ作業ができないということを意味する。即ち、デバッグ作業時においては、マイクロコンピュータのクロック周波数を下げなければいけなくなる。

【0087】

このような問題を解決するために本実施形態では、図16に示すように、マイクロコンピュータ140とデバッグツール150とで、サンプリングクロックを生成するためのBCLKを共有させている。

【0088】

より具体的には、マイクロコンピュータ140（第1の情報処理装置）は、通信部142（図3のS1048に相当）を含む。そして通信部142は、送受信回路144（図6の送受信バッファ70、シフトレジスタ76及び送受信切替部78に相当）、分周回路146（図6の分周回路82に相当）を含む。この分周回路146は、図17（A）に示すように、BCLK（第1のクロック）を分周し、調歩同期式で送受信されるデータの各ビットをサンプリングするためのサンプリングクロックSMC1を生成する。そして、送受信回路144は、このSMC1に基づいてデータを送受信する。更にマイクロコンピュータ140は、BCLKをデバッグツール150に供給する。

【0089】

デバッグツール150（第2の情報処理装置）は、通信部152を含む。そして通信部152は、送受信回路154（図7のCPU90、送受信切替部92に相当）、分周回路156（図7の分周回路96に相当）を含む。この分周回路156は、図17（B）に示すように、マイクロコンピュータ140から供給されたBCLKを分周し、サンプリングクロックSMC2を生成する。そして、送受信回路154は、このSMC2に基づいてデータを送受信する。

【0090】

このように本実施形態では、調歩同期式でありながら、サンプリングクロックSMC1、SMC2を生成するためのBCLKをマイクロコンピュータ140とデバッグツール150とで共有する。これにより、通信データのサンプリングエラーの発生率を、図14（B）に示す一般的な調歩同期式通信に比べて格段に減少できる。また、図14（A）の同期式通信では、4本の通信ラインが必要であったが、本実施形態では図16に示すように2本の通信ラインで済む（全2重の場合には3本）。したがって、マイクロコンピュータ140とデバッグツール150との間の通信ラインの本数を、図14（A）に比べて少なくできる。この結果、マイクロコンピュータ140の端子数を減らすことが可能になり、マイクロコンピュータ140の低コスト化を図れるようになる。

【0091】

特に、図14（B）の構成では、CLK1（及びCLK2）の周波数が高くなればなるほど、通信データのサンプリングエラーの発生率も高くなる。このため、デバッグ作業時においてマイクロコンピュータ340のクロック周波数を高くできず、マイクロコンピュータ340が高速動作する環境でのデバッグ作業ができなかった。

【009-2】

これに対して、図16の本実施形態では、マイクロコンピュータ140もデバッガツール150もBCLKに基づいてサンプリングクロックを生成している。このため、BCLKの周波数が高くなってしまっても、サンプリングエラーの発生率がそれほど高くならない。この結果、マイクロコンピュータ140が高速動作する環境でのデバッグ作業が可能になり、より実動作時に近い環境でのデバッグ作業が可能になる。

【009-3】

更に本実施形態では、図18に示すように、通信部142に分周比制御部148（図6の制御レジスタ84に相当）を含ませると共に、通信部152に分周比制御部158（図7の制御レジスタ98に相当）、周波数検出回路159（図7の周波数検出回路95に相当）を含ませている。これにより、SMC1を生成する際の分周比FD1と、SMC2を生成する際の分周比FD2を可変に制御することが可能になる。この結果、BCLKの周波数が変化しても、最適で高速な通信速度でデータを通信することが可能になる。

【009-4】

即ち図19（A）に示すようにBCLKの周波数が低くなった場合には、図17（A）、（B）で16であった分周比FD1、FD2が例えば8に変更される。これにより、サンプリングクロックSMC1、SMC2は、BCLKを16分周したクロックから8分周したクロックに変更される。この結果、1ビットのデータに対応するBCLKのクロック数が、16であったものが（16クロックモード）、8に変更される（8クロックモード）。

【009-5】

また図19（B）に示すようにBCLKの周波数が更に低くなった場合には、

図19 (A) で8であった分周比FD1、FD2が例えば4に変更される。これにより、サンプリングクロックSMC1、SMC2は、BCLKを8分周したクロックから4分周したクロックに変更される。この結果、1ビットのデータに対応するBCLKのクロック数が、8であったものが（8クロックモード）、4になるように変更される（4クロックモード）。

【009-6】

このようにすることで、BCLKの周波数が低くなつた場合にも、データの通信速度は結局低くならなくなる。この結果、最適で高速な通信速度でデータを通信することが可能になる。

【009-7】

特に、マイクロコンピュータのクロック周波数は、マイクロコンピュータを使用するユーザ毎に異なつたものになるのが一般的である。即ち、あるユーザは60MHZのクロックでマイクロコンピュータを動作させ、別のユーザは20MHZのクロックでマイクロコンピュータを動作させる。

【009-8】

しかしながら、図14 (A)、(B) の通信方式では、マイクロコンピュータのクロック周波数が変化すると、データの通信速度も変化してしまう。即ち、クロック周波数が低くなるとデータの通信速度も低くなつてしまつ。したがつて、最大限の通信速度でデータを通信することができなくなる。

【009-9】

これに対して、本実施形態では、マイクロコンピュータのクロック周波数が、マイクロコンピュータを使用するユーザに応じて変化すると、分周比FD1、FD2も変化し、1ビットのデータに対応するクロック数も変化する。即ちクロック周波数が低くなると、分周比FD1、FD2も小さくなり、1ビットのデータに対応するクロック数も少なくなる。この結果、通信速度は結局低くならず、最適で高速な通信速度で通信を行えるようになる。即ち、多様な周波数のクロックを使用するユーザに応じて、最適な通信速度で通信を行えるようになる。

【010-0】

次に、通信部142、152での処理について図20、図21のフローチャー

トを用いて更に詳細に説明する。

【0101】

図20に示すように、まず、図18の通信部152内の周波数検出回路159が、マイクロコンピュータ140から供給されるBCLKの周波数を検出する（ステップV1）。そして、BCLKの周波数が30MHz以上であるか否かを判断し（ステップV2）、30MHz以上である場合には、分周比制御部158が分周比FD2を16に設定する（ステップV3）。そして、FD2が16であることを知らせる分周比データを、送受信回路154を介してマイクロコンピュータ140に送信する（ステップV4）。次に、分周回路156が $FD2 = 16$ でBCLKを分周しSMC2を生成する（ステップV5）。そして、以降は、このSMC2によりデータの送受信を行う。

【0102】

BCLKの周波数が30MHzより低い場合には、周波数が15MHz以上か否かを判断する（ステップV6）。そして、周波数が15MHz以上である場合には $FD2 = 8$ に設定し（ステップV7）、ステップV4、V5と同様の処理を行う（ステップV8、V9）。また、BCLKの周波数が15MHzより低い場合には $FD2 = 4$ に設定し（ステップV10）、ステップV4、V5と同様の処理を行う（ステップV11、V12）。

【0103】

一方、マイクロコンピュータ140側では図21のフローチャートに示すような処理を行う。即ち、まず、デバッグツール150から送受信回路144を介して分周比データを受信する（ステップW1）。次に、受信した分周比データに基づいて分周比制御部148が分周比FD1を決定する（ステップW2）。そして、分周回路146は、このFD1でBCLKを分周しSMC1を生成する。そして、以降は、このSMC1でデータの送受信を行う。

【0104】

1.1. ミニモニタ部の詳細な処理例

次にミニモニタ部の詳細な処理例について説明する。

【0105】

図22に示すように、ユーザプログラムの実行中にブレークが発生すると、ミニモニタプログラムの処理がスタートし、ユーザプログラム実行モードからデバッグモードに移行する。そして、ミニモニタプログラムが所与のコマンド処理を行いリターン命令を実行すると、デバッグモードからユーザプログラム実行モードに戻る。

【0106】

図23、図24に、デバッグモードでのミニモニタプログラムの処理を表すフローチャートを示す。

【0107】

デバッグモードに移行すると、ミニモニタプログラムは、まず、図3のCPU22の内部レジスタ24の内容をミニモニタRAM44に退避する（ステップS1）。そして、ミニモニタプログラムが使用する制御レジスタ46の設定処理を行う（ステップS2）。

【0108】

次に、デバッグツール60から受信した14バイトのデータ（図10（A）参照）を、受信バッファ74（図6参照）にライトする（ステップS3）。そして、受信バッファ74のデータの先頭の1バイト（コマンド識別データID）をチェックする（ステップS4）。

【0109】

そして図24に示すように、IDがリードコマンドを示すものであった場合には（図10（D）参照）、受信バッファ74からリードアドレスを取得する（ステップS5、S6）。そして、取得したリードアドレスからデータをリードし、送信バッファ72にライトする（ステップS7）。次に、送信バッファ72のデータをデバッグツール60に送信する（ステップS8）。そして、図23のステップS3に戻り、次の受信データを受信バッファ74にライトする。

【0110】

IDがライトコマンドを示すものであった場合には（図10（C）参照）、受信バッファ74からライトアドレスを取得する（ステップS9、S10）。そして、受信バッファ74からライトデータを取得し、ステップS10で取得したラ

イトアドレスにライトする（ステップS11）。

【0111】

IDが外部ルーチンジャンプコマンドを示すものであった場合には（図11（A）参照）、受信バッファ74からルーチンアドレスを取得する（ステップS12、S13）。そして、外部ルーチンにジャンプ後、ミニモニタプログラムにリターンする（ステップS14）。

【0112】

IDがGOコマンドを示すものであった場合には（図11（B）参照）、ミニモニタRAM44に退避したデータを内部レジスタ24にリストアする（ステップS15、S16）。そして、図22に示すようにユーザプログラムにリターンし、デバッグモードから抜ける（ステップS17）。

【0113】

一方、IDがリード、ライト、外部ルーチンジャンプ、GOコマンドのいずれでもなかった場合には、処理が必要ないと判断する（ステップS15、S18）。そして送信バッファ72にダミーのデータをライトする（ステップS19）。なお、図24では、データフィルコマンドの処理については省略している。

【0114】

以上のようにして、デバッグコマンドを変換することで得られたプリミティブコマンドが、ミニモニタプログラムにより実行されることになる。

【0115】

12. 電子機器

次に、以上の本実施形態のマイクロコンピュータを含む電子機器について説明する。

【0116】

例えば図25（A）に電子機器の1つであるカーナビゲーションシステムの内部ブロック図を示し、図26（A）にその外観図を示す。カーナビゲーションシステムの操作はリモコン510を用いて行われ、GPSやジャイロからの情報に基づいて位置検出部520が車の位置を検出する。地図などの情報はCDROM530（情報記憶媒体）に格納されている。画像メモリ540は画像処理の際の

作業領域になるメモリであり、生成された画像は画像出力部550を用いて運転者に表示される。マイクロコンピュータ500は、リモコン510、位置検出部520、CDROM530などのデータ入力源からデータを入力し、種々の処理を行い、処理後のデータを画像出力部550などの出力装置を用いて出力する。

【0117】

図25(B)に電子機器の1つであるゲーム装置の内部ブロック図を示し、図26(B)にその外観図を示す。このゲーム装置では、ゲームコントローラ560からのプレーヤの操作情報、CDROM570からのゲームプログラム、ICカード580からのプレーヤ情報等に基づいて、画像メモリ590を作業領域としてゲーム画像やゲーム音を生成し、画像出力部610、音出力部600を用いて出力する。

【0118】

図25(C)に電子機器の1つであるプリンタの内部ブロック図を示し、図26(C)にその外観図を示す。このプリンタでは、操作パネル620からの操作情報、コードメモリ630及びフォントメモリ640から文字情報に基づいて、ビットマップメモリ650を作業領域として、印刷画像を生成し、プリント出力部660を用いて出力する。またプリンタの状態やモードを表示パネル670を用いてユーザに伝える。

【0119】

本実施形態のマイクロコンピュータ又はデバッグシステムによれば、図25(A)～図26(C)の電子機器を動作させるユーザプログラムの開発の容易化、開発期間の短縮化を図れるようになる。またマイクロコンピュータが実動作する環境と同じ環境で、ユーザプログラムのデバッグ作業を行うことができるため、電子機器の信頼性を高めることができる。また電子機器に組み込まれるマイクロコンピュータのハードウェアを小規模化、低コスト化できるため、電子機器の低コスト化も図れるようになる。更にミニモニタプログラムの命令コードサイズは小さいため、ユーザがプログラムや各種データの格納に使用するメモリ領域を最大限に確保でき、場合によれば、ユーザーが使用する該メモリ領域を全く使用せずにモニタプログラムを載せることができる。

【0120】

なお本実施形態のマイクロコンピュータを適用できる電子機器としては、上記以外にも例えば、携帯電話（セルラーフォン）、PHS、ページャ、オーディオ機器、電子手帳、電子卓上計算機、POS端末、タッチパネルを備えた装置、プロジェクト、ワードプロセッサ、パーソナルコンピュータ、テレビ、ビューファインダ型又はモニタ直視型のビデオテープレコーダなど種々のものを考えることができる。

【0121】

なお、本発明は本実施形態に限定されず、本発明の要旨の範囲内で種々の変形実施が可能である。

【0122】

例えば本発明のプリミティブコマンドとしては、本実施形態で説明したものが特に望ましいが、これに限定されるものではない。

【0123】

またマイクロコンピュータやミニモニタ部（第1のモニタ手段）の構成も本実施形態で説明したものに限定されず、種々の変形実施が可能である。

【0124】

またデバッグシステムの構成も図7に示すものに限られるものではない。

【0125】

【図面の簡単な説明】

【図1】

図1（A）はCPU置き換え型のICE、図1（B）はモニタプログラム実装型ICEの例を示す図である。

【図2】

本実施形態の特徴について説明するための図である。

【図3】

本実施形態のマイクロコンピュータ、デバッグシステムの構成例を示す機能ブロック図である。

【図4】

デバッグモード時のメモリマップを示す図である。

【図5】

図5 (A)、(B)、(C)、(D)は、デバッグコマンドをプリミティブコマンドへ変換(分解)する処理について説明するための図である。

【図6】

S-I-Oの構成例を示す機能ブロック図である。

【図7】

デバッグツールの構成例を示す機能ブロック図である。

【図8】

図8 (A)、(B)は、リアルタイムトレース処理について説明するための図である。

【図9】

図9 (A)、(B)、(C)は、ミニモニタ部とメインモニタ部との間の通信手法について説明するための図である。

【図10】

図10 (A)、(B)、(C)、(D)は、送受信データのフォーマット及び種類について説明するための図である。

【図11】

図11 (A)、(B)も、送受信データのフォーマット及び種類について説明するための図である。

【図12】

図12 (A)、(B)は、送受信データが可変長、固定長の場合のミニモニタプログラムのソースコードのサイズについて説明するための図である。

【図13】

図13 (A)、(B)は、ミニモニタプログラムをROMに格納する手法について説明するための図である。

【図14】

図14 (A)、(B)は、同期式、調歩同期式の通信手法について説明するための図である。

【図15】

図15 (A)、(B)は、一般的な調歩同期式でのクロックとサンプリングクロックとサンプリングデータのタイミング波形を示す図である。

【図16】

本実施形態の通信手法について説明するための図である。

【図17】

図17 (A)、(B)は、図16の手法でのクロックとサンプリングクロックとサンプリングデータのタイミング波形を示す図である。

【図18】

本実施形態の通信手法について説明するための図である。

【図19】

図19 (A)、(B)は、図18の手法でのクロックとサンプリングクロックとサンプリングデータのタイミング波形を示す図である。

【図20】

デバッグツール側での分周比設定処理について説明するためのフローチャートである。

【図21】

マイクロコンピュータ側での分周比設定処理について説明するためのフローチャートである。

【図22】

ユーザプログラム実行モードからデバッグモードへの移行について説明するための図である。

【図23】

本実施形態の詳細な処理例を説明するためのフローチャートである。

【図24】

本実施形態の詳細な処理例を説明するためのフローチャートである。

【図25】

図25 (A)、(B)、(C)は、種々の電子機器の内部ブロック図の例である。

【図26】

図26 (A)、(B)、(C) は、種々の電子機器の外観図の例である。

【符号の説明】

1 0 マイクロコンピュータ
 1 2 C P U (中央処理ユニット)
 1 4 ミニモニタ部 (第1のモニタ手段)
 1 6 メインモニタ部 (第2のモニタ手段)
 2 0 マイクロコンピュータ
 2 2 C P U (中央処理ユニット)
 2 4 内部レジスタ
 2 6 B C U (バス制御ユニット)
 2 8 内部メモリ
 3 0 クロック生成部
 3 1 バス
 3 2 バス
 3 3 外部バス
 3 4 内部バス
 3 6 外部メモリ
 4 0 ミニモニタ部 (第1のモニタ手段)
 4 2 ミニモニタ R O M
 4 4 ミニモニタ R A M
 4 6 個別レジスタ
 4 8 S I O
 5 0 トレース部
 6 0 デバッグツール
 6 2 メインモニタ部 (第2のモニタ手段)
 6 6 ホストシステム
 7 0 送受信バッファ
 7 2 送信バッファ

74 受信バッファ

76 シフトレジスタ

78 送受信切替部

80 クロック制御部

82 分周回路

84 制御レジスタ

90 CPU

92 送受信切替部

94 クロック制御部

95 周波数検出部

96 分周回路

98 制御レジスタ

100 アドレスアップカウンタ

102 セレクタ

104 トレースメモリ

106 セレクタ

108 ROM

110 メインモニタROM

112 RAM

114 RS232Cインターフェース

116 パラレルインターフェース

118 クロック生成部

140 マイクロコンピュータ（第1の情報処理装置）

142 通信部

144 送受信回路

146 分周回路

148 分周比制御部

150 デバッグツール（第2の情報処理装置）

152 通信部

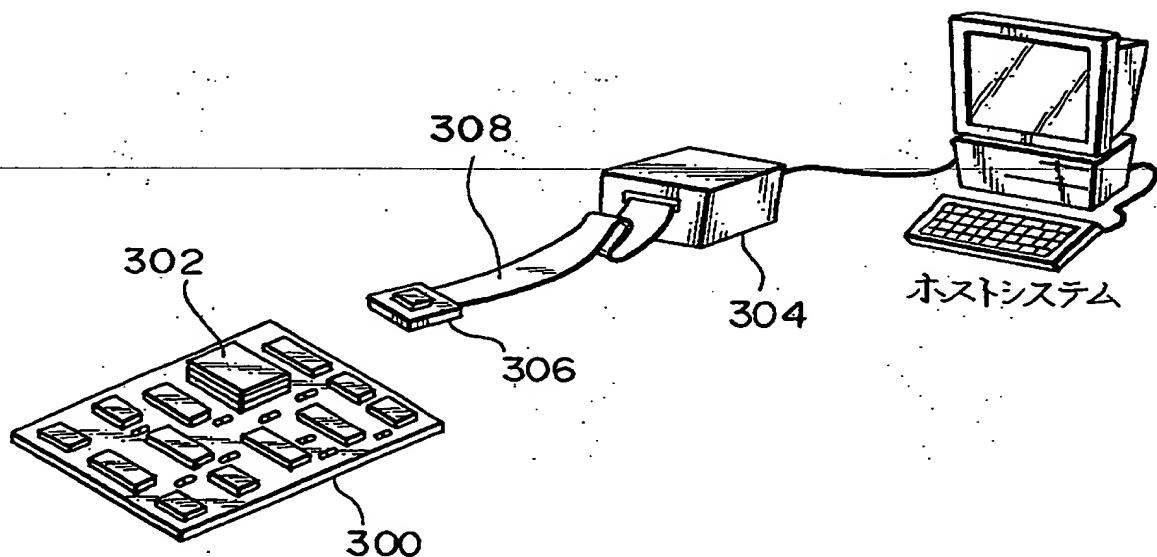
- 1 5 4 送受信回路
- 1 5 6 分周回路
- 1 5 8 分周比制御部
- 1 5 9 周波数検出回路

【書類名】

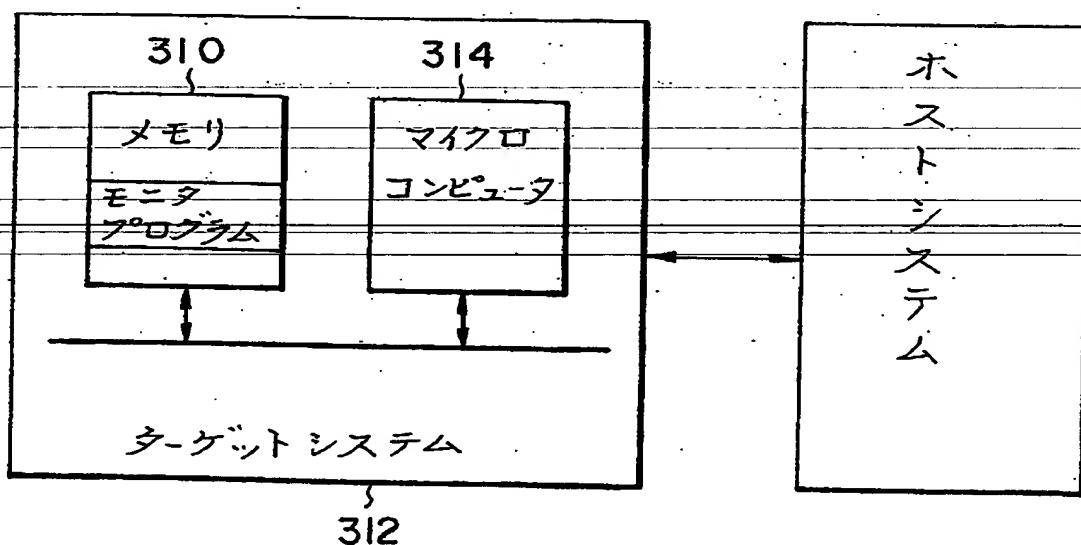
図面

【図1】

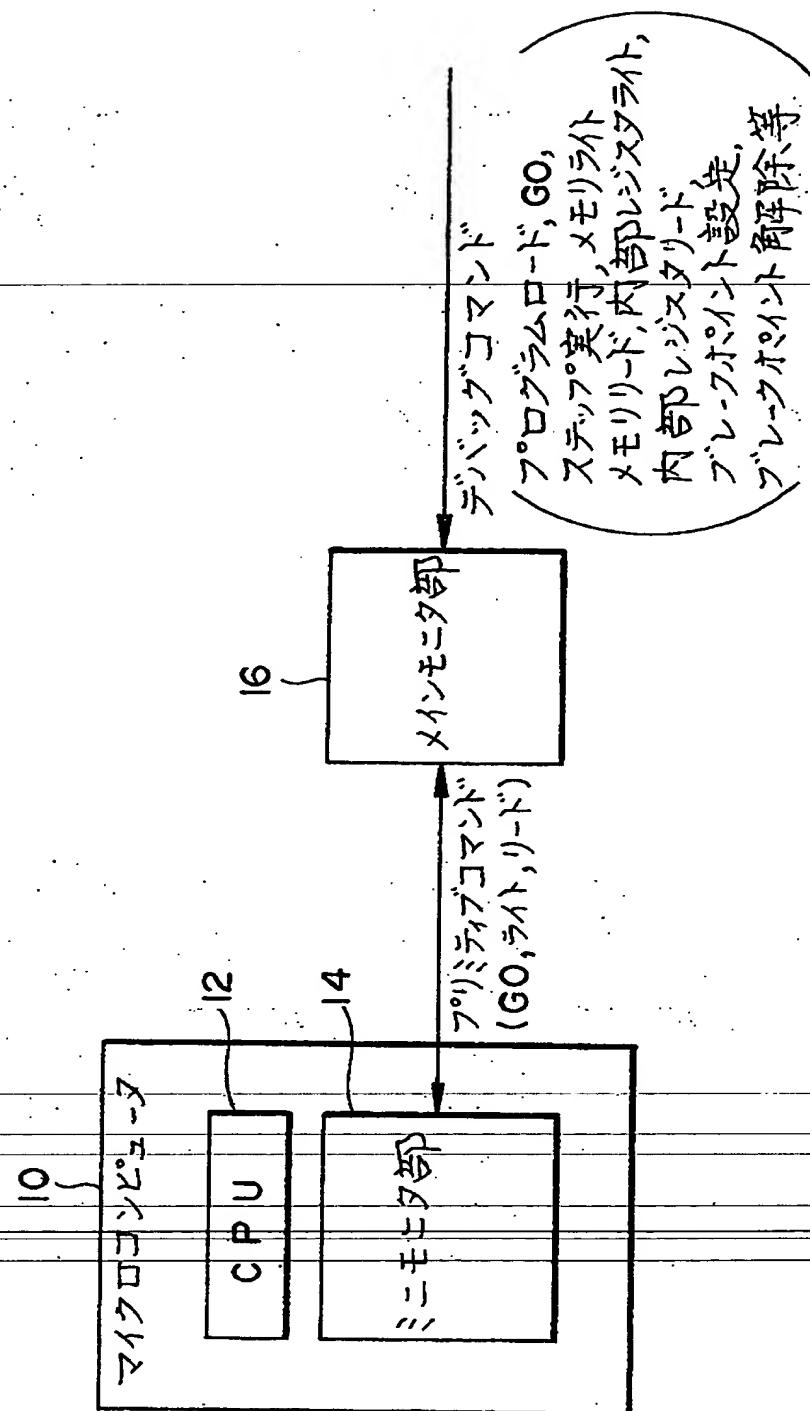
(A)



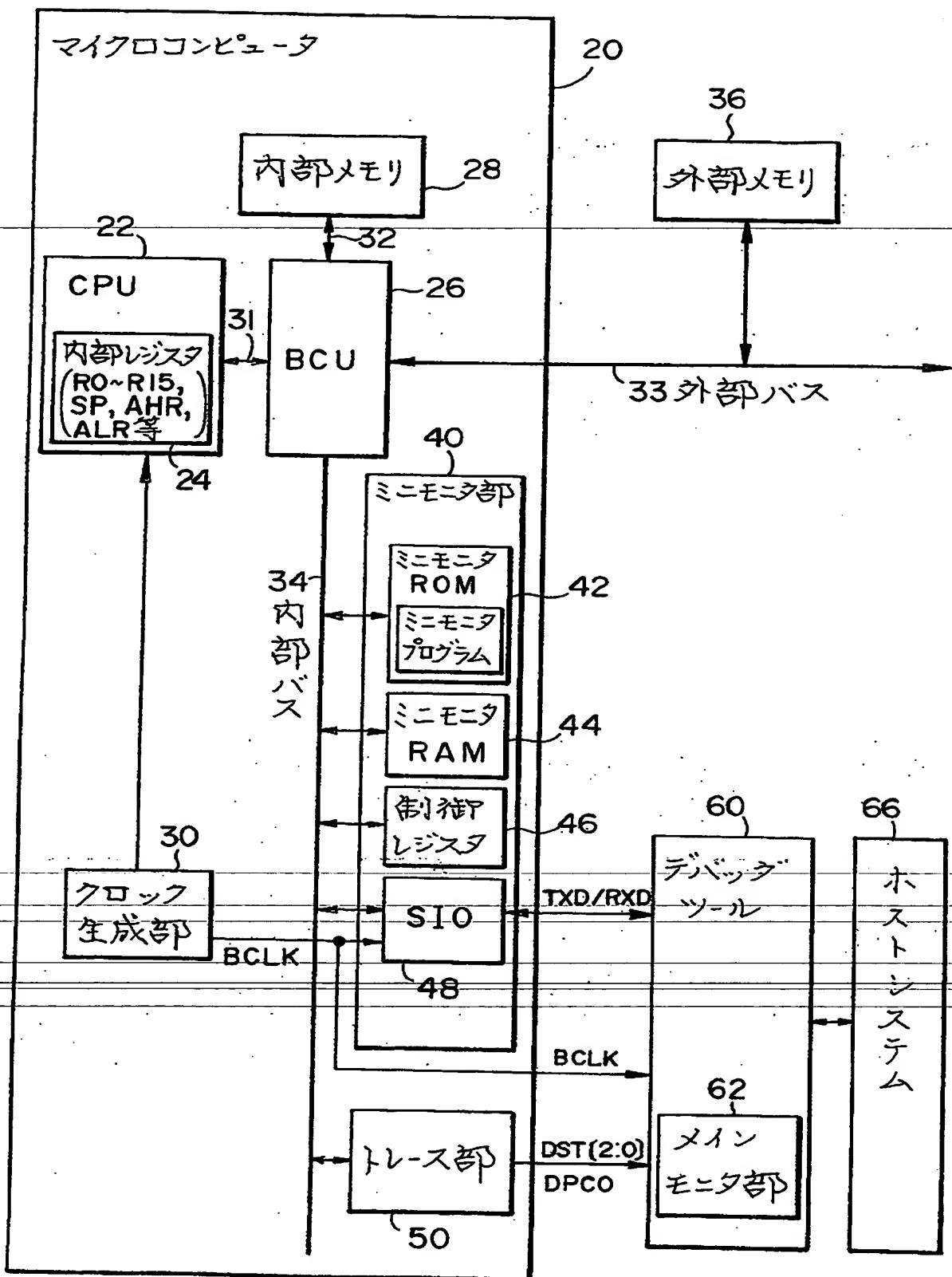
(B)



【図2】

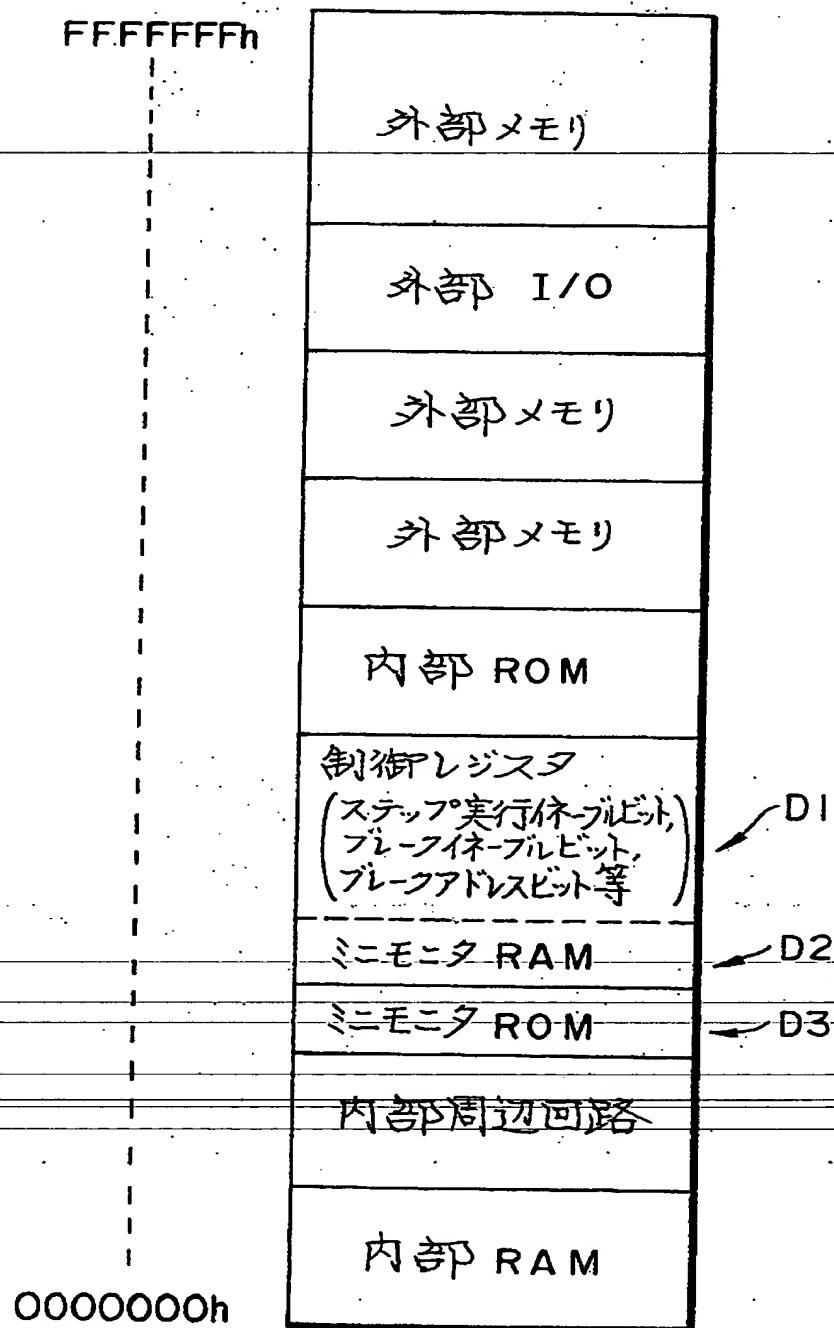


【図3】



【図4】

デバッグモード時のメモリマップ



【図5】

(A) プログラムコード

$$\left(\begin{array}{l} 80010h, 12バイト, ADD\cdots, SUB\cdots \\ (AND\cdots, OR\cdots, XOR\cdots, LD.W\cdots) \end{array} \right) \Rightarrow \begin{array}{l} \text{ライト}(80010h, ADD\cdots, SUB\cdots) \\ +\text{ライト}(80014h, ADD\cdots, OR\cdots) \\ +\text{ライト}(80018h, XOR\cdots, LD.W\cdots) \end{array}$$

(B) ステップ実行

\Rightarrow 割り振アレジスタのステップ実行イネーブル
ビットへのライト
+GO

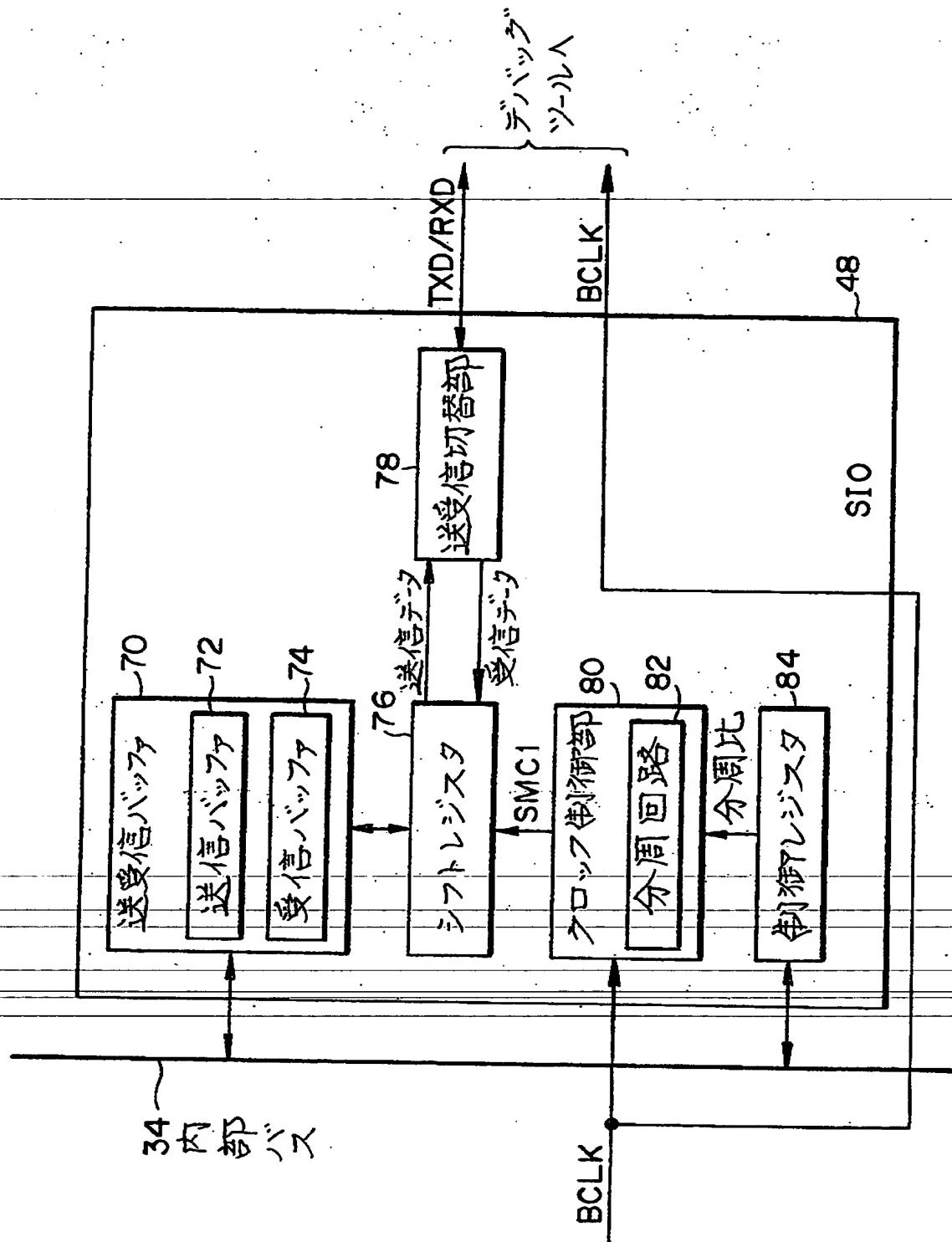
(C) 内部レジスタリード

\Rightarrow メモリマップ上のモニタRAMへのリード

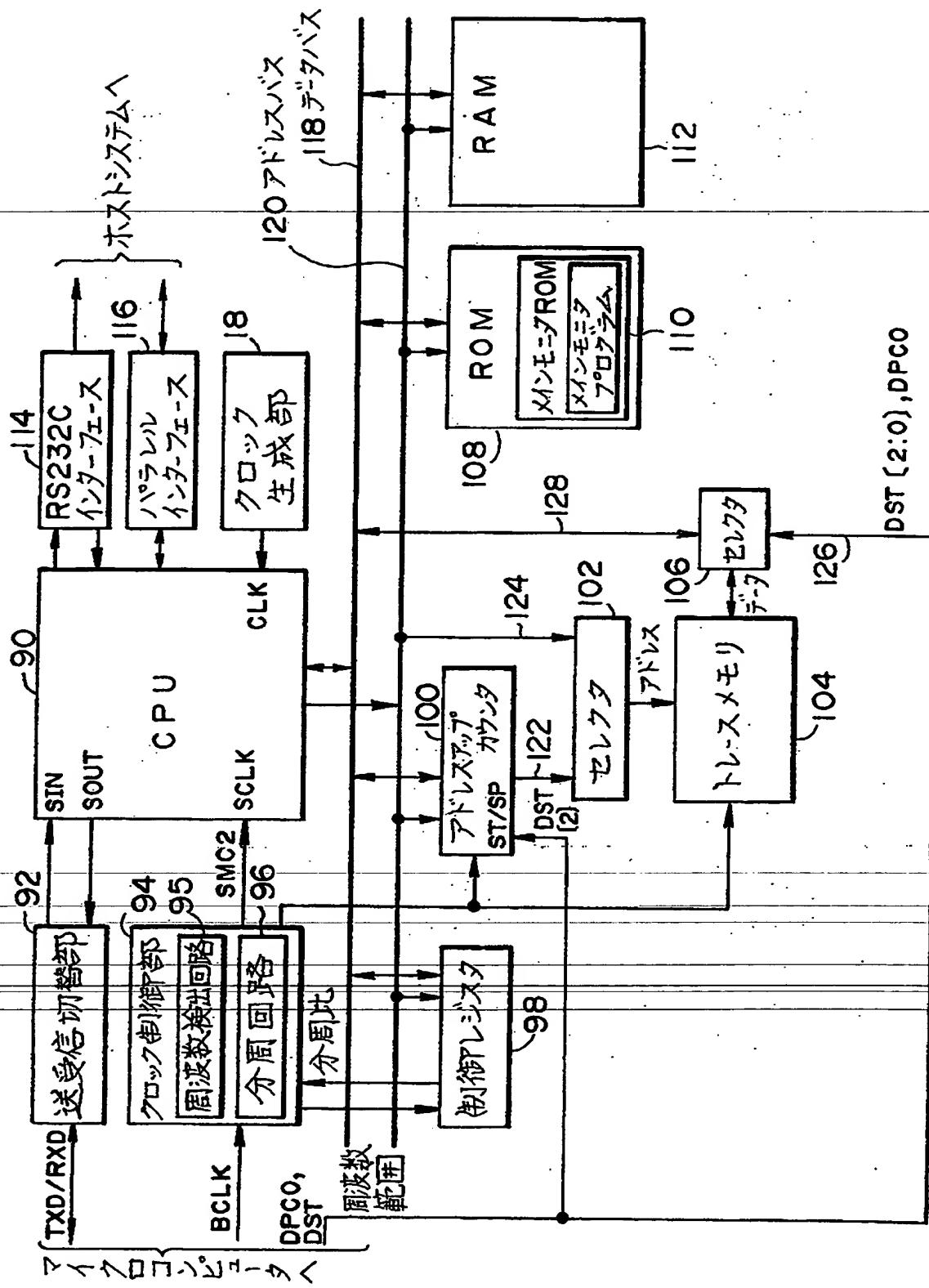
(D) ブレークポイント設定

\Rightarrow 割り振アレジスタのブレークポイントリセット
及びブレークアドレスセットへのライト

【図6】

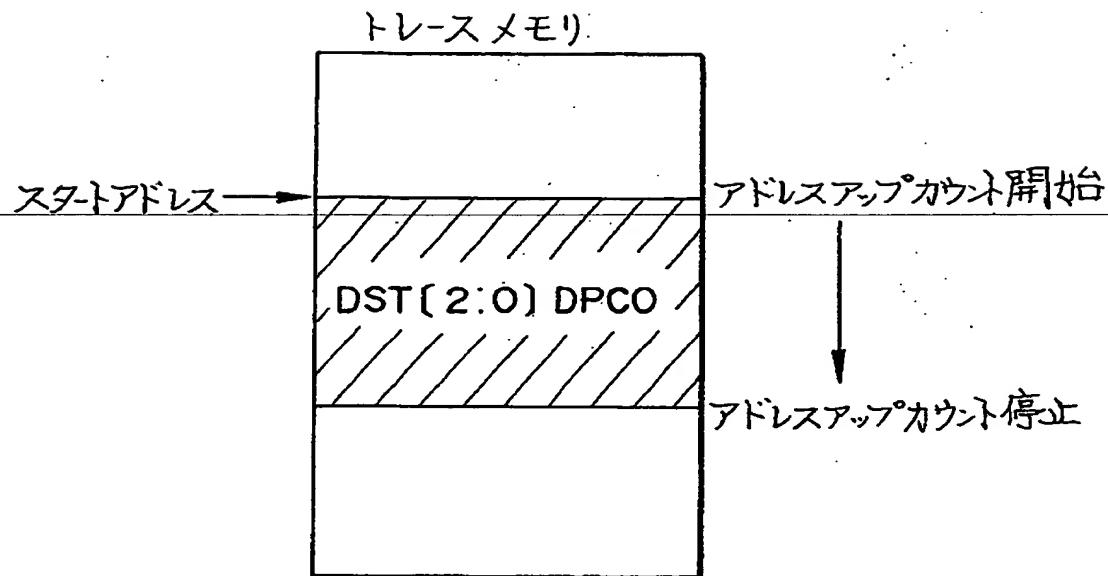


【図7】

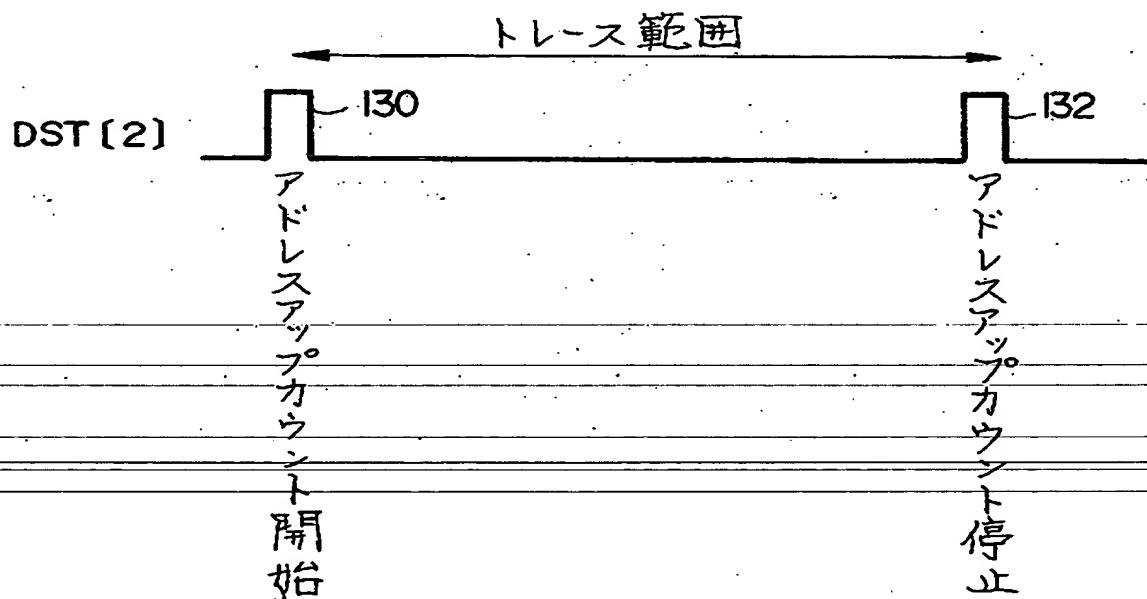


【図8】

(A)

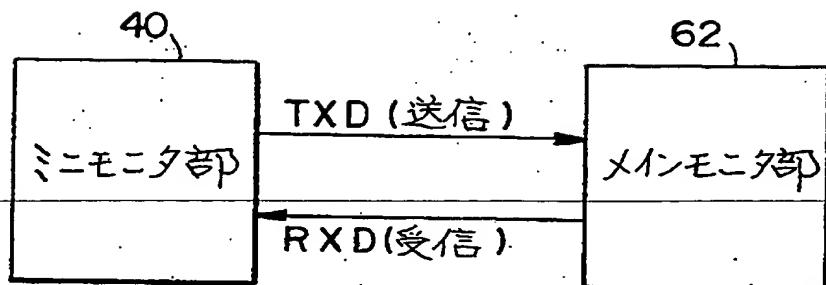


(B)

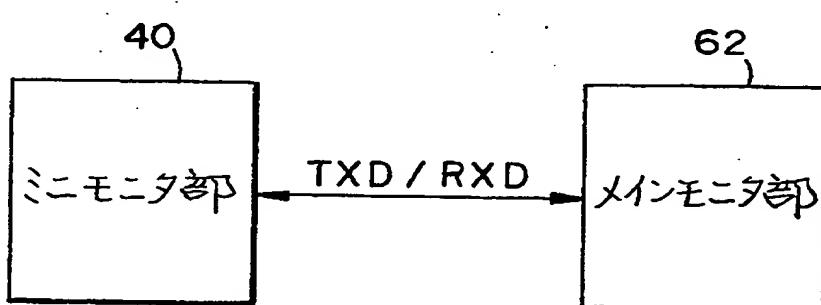


【図9】

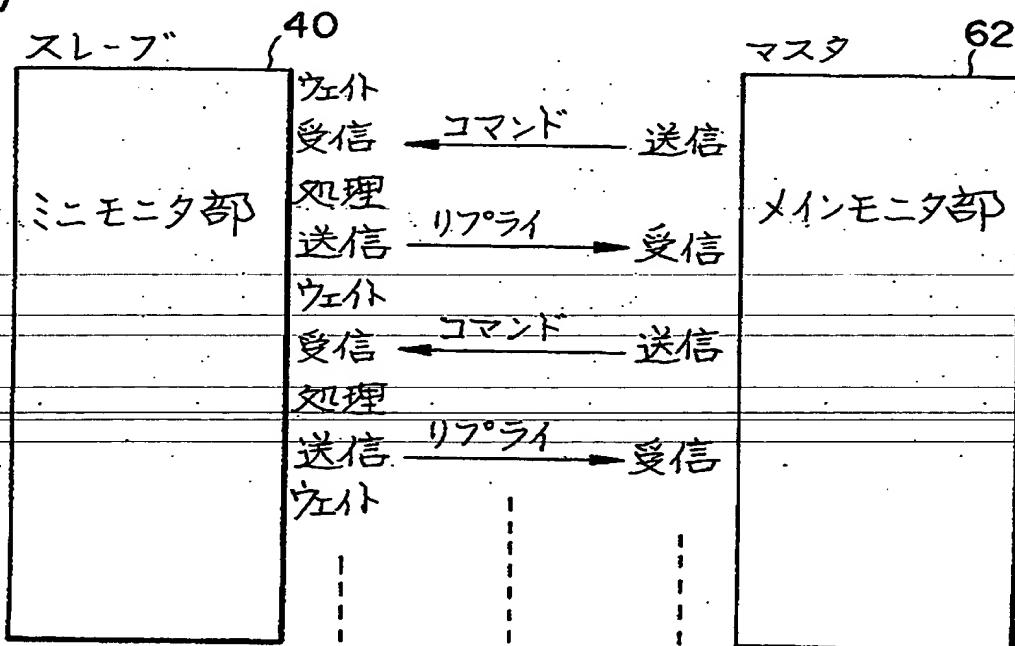
(A)



(B)



(C)



【図10】

(A) データフォーマット

ID	1バイト
データサイズ	1バイト
アドレス	4バイト
データ1	4バイト
データ2	4バイト

14バイトの固定長

(B) GO コマンド

受信データ	送信データ
00h	無し。実行に移る
—	
—	
—	
—	

(C) ライトコマンド

受信データ	送信データ
01h	—
—	—
ライトアドレス	—
ライトデータ1	01h
ライトデータ2	—

(D) リードコマンド

受信データ	送信データ
02h	—
—	—
リードアドレス	—
—	リードデータ1
—	リードデータ2

【図11】

(A) 外部ルーチンジャンプコマンド

受信データ	送信データ
03h	—
18h	—
%R12	—
%R13	—
%R14	(戻り値。0が正常終了)

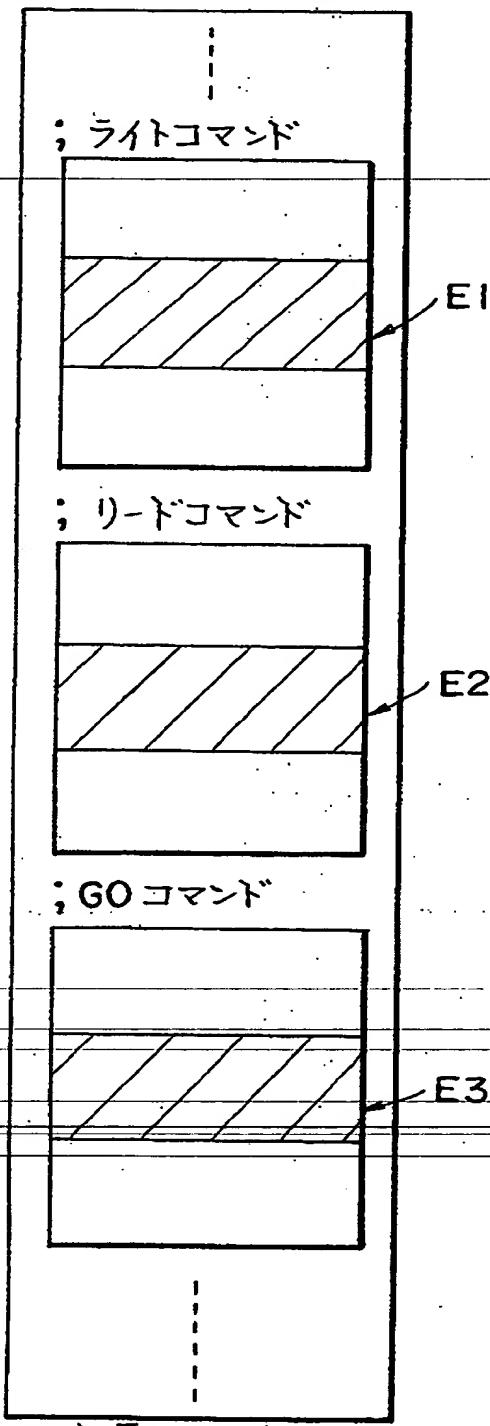
(B) データファイルコマンド

受信データ	送信データ
04h	—
1,2,4	—
スタートアドレス	—
ファイル回数	—
ファイルパターン	04h

【図12】

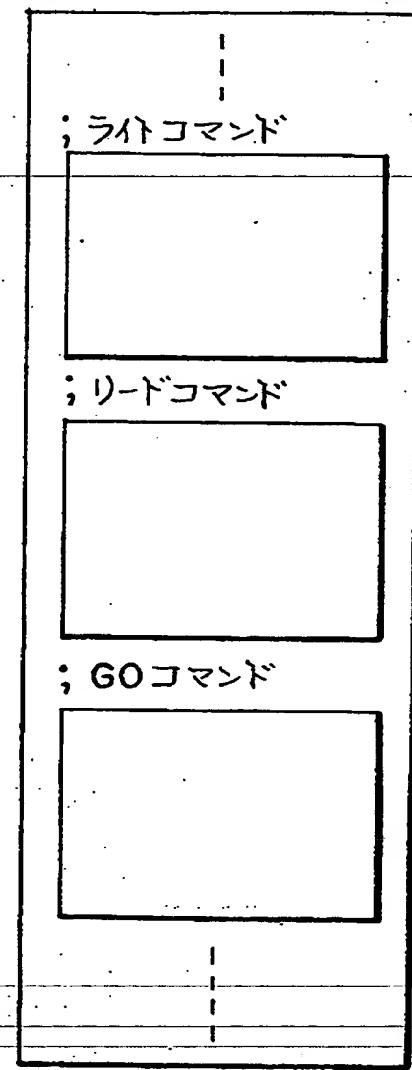
(A)

ミニモニタプログラムのソースコード



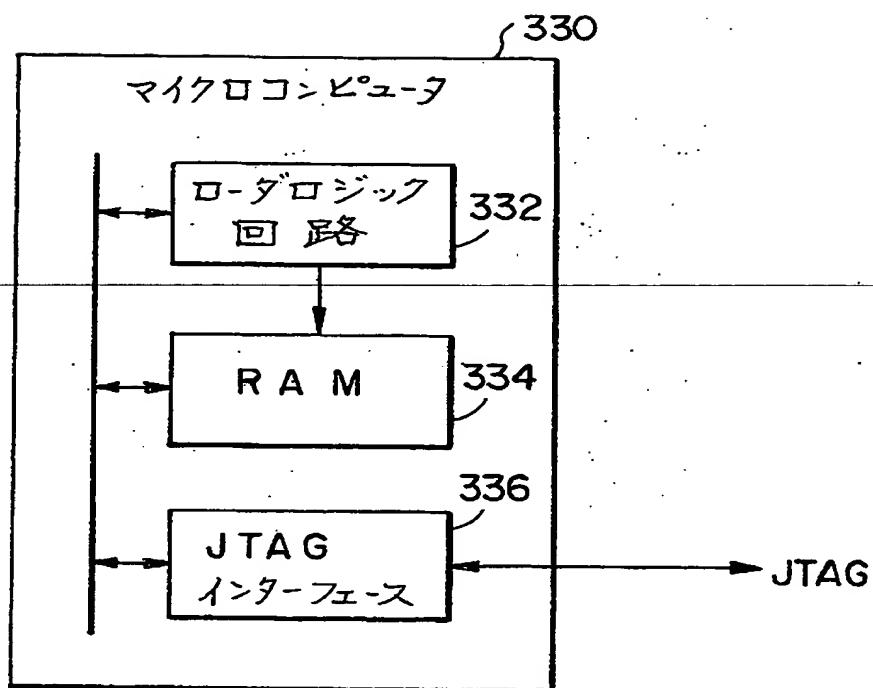
(B)

ミニモニタプログラムのソースコード

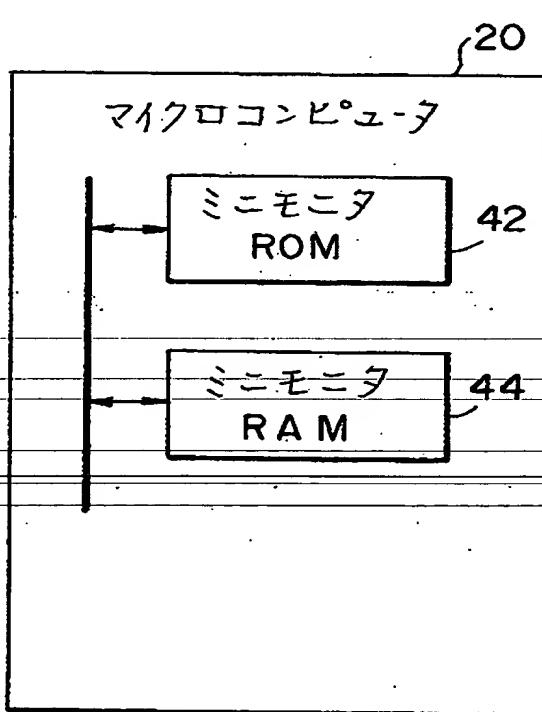


【図13】

(A)

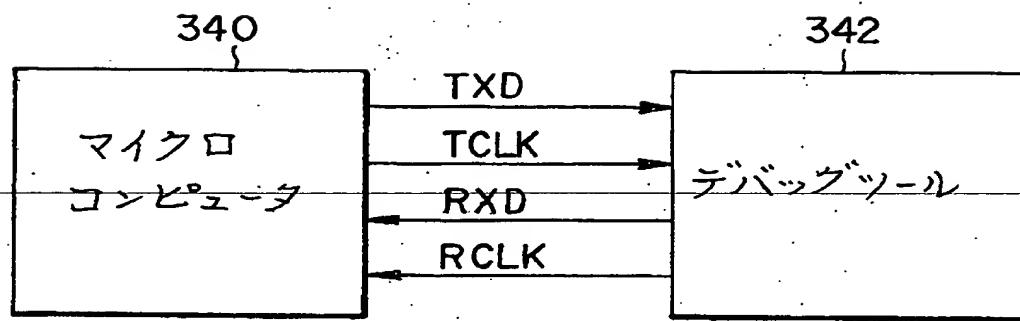


(B)

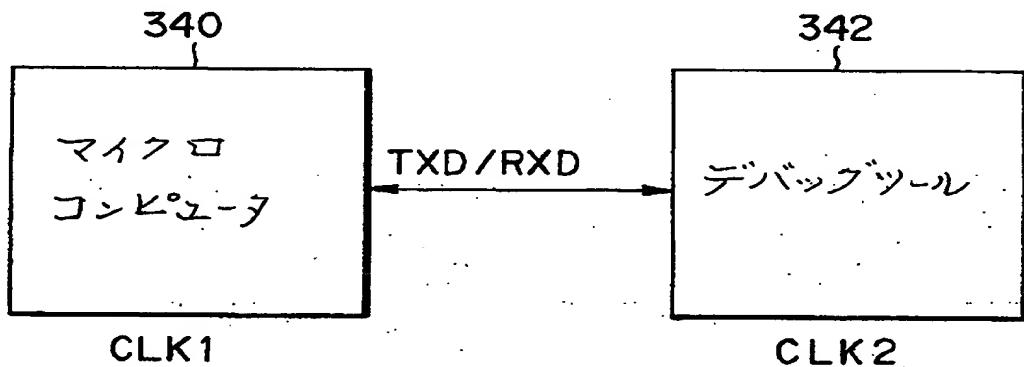


【図14】

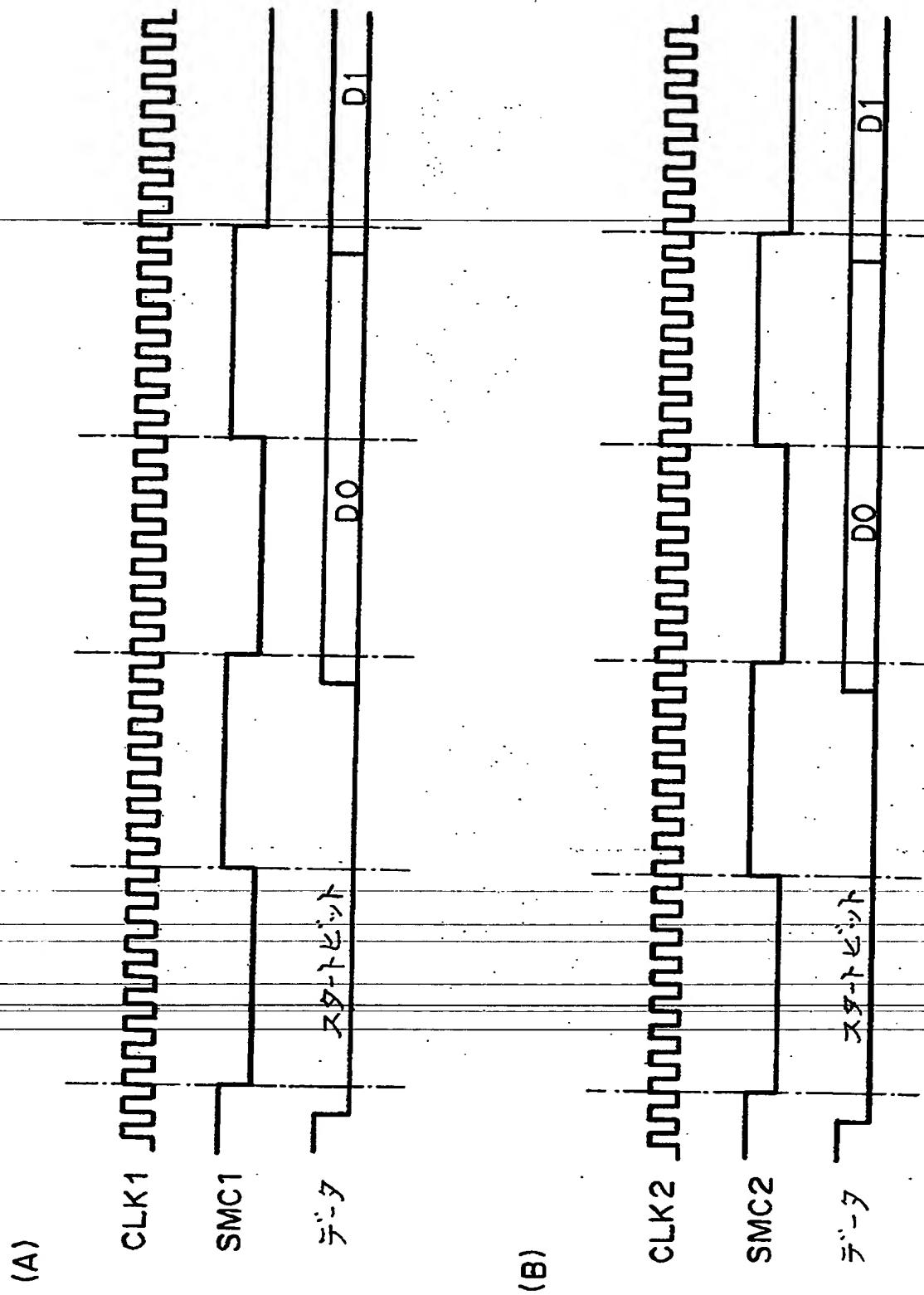
(A)



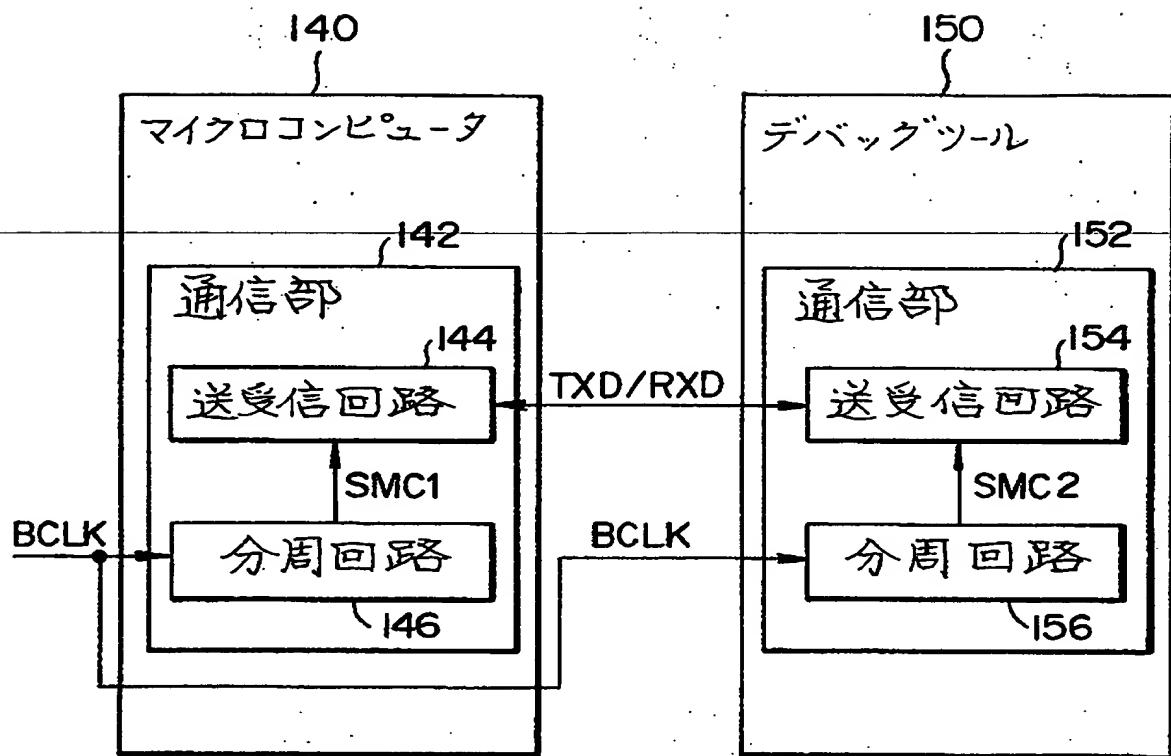
(B)



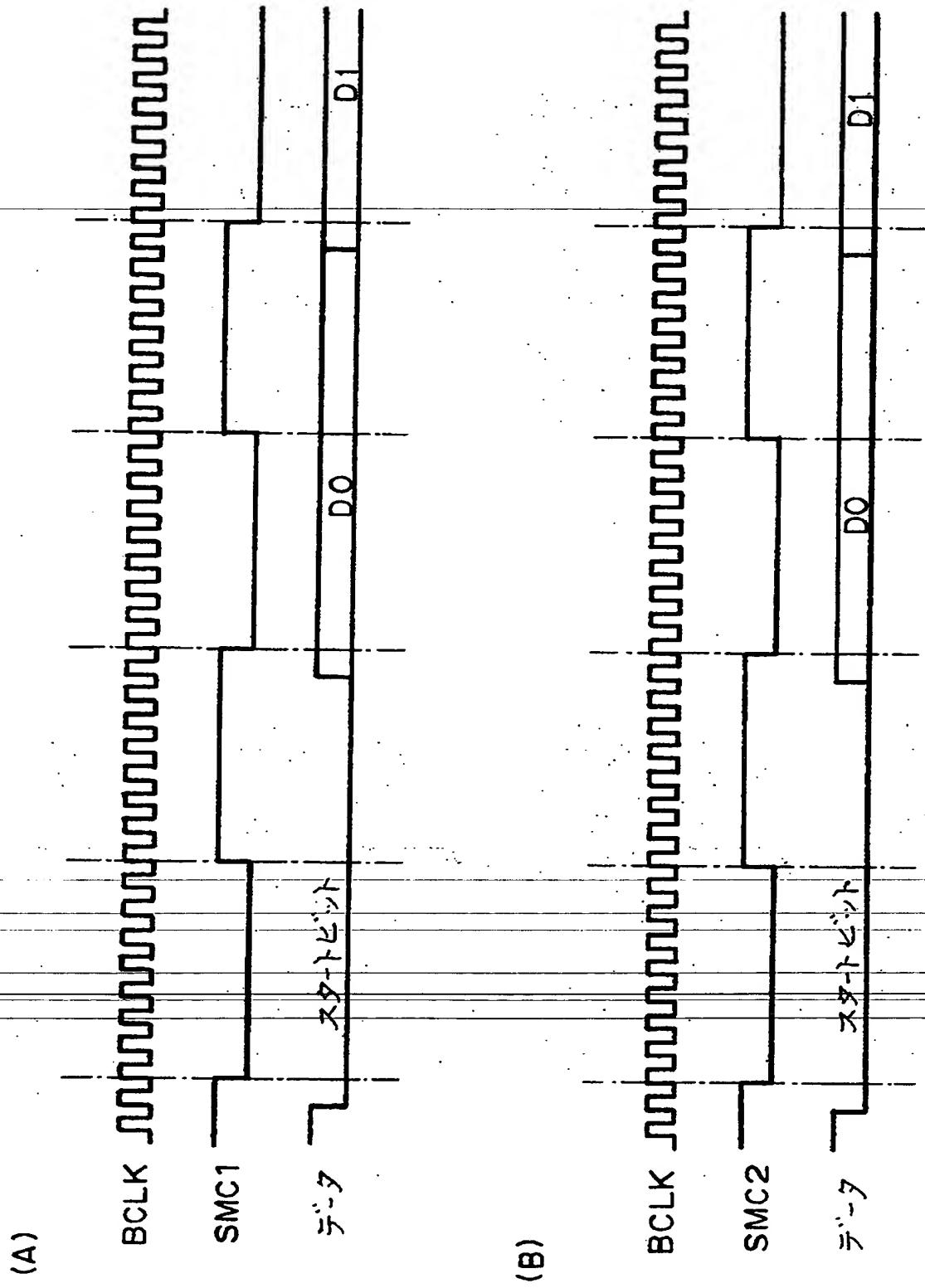
【図15】



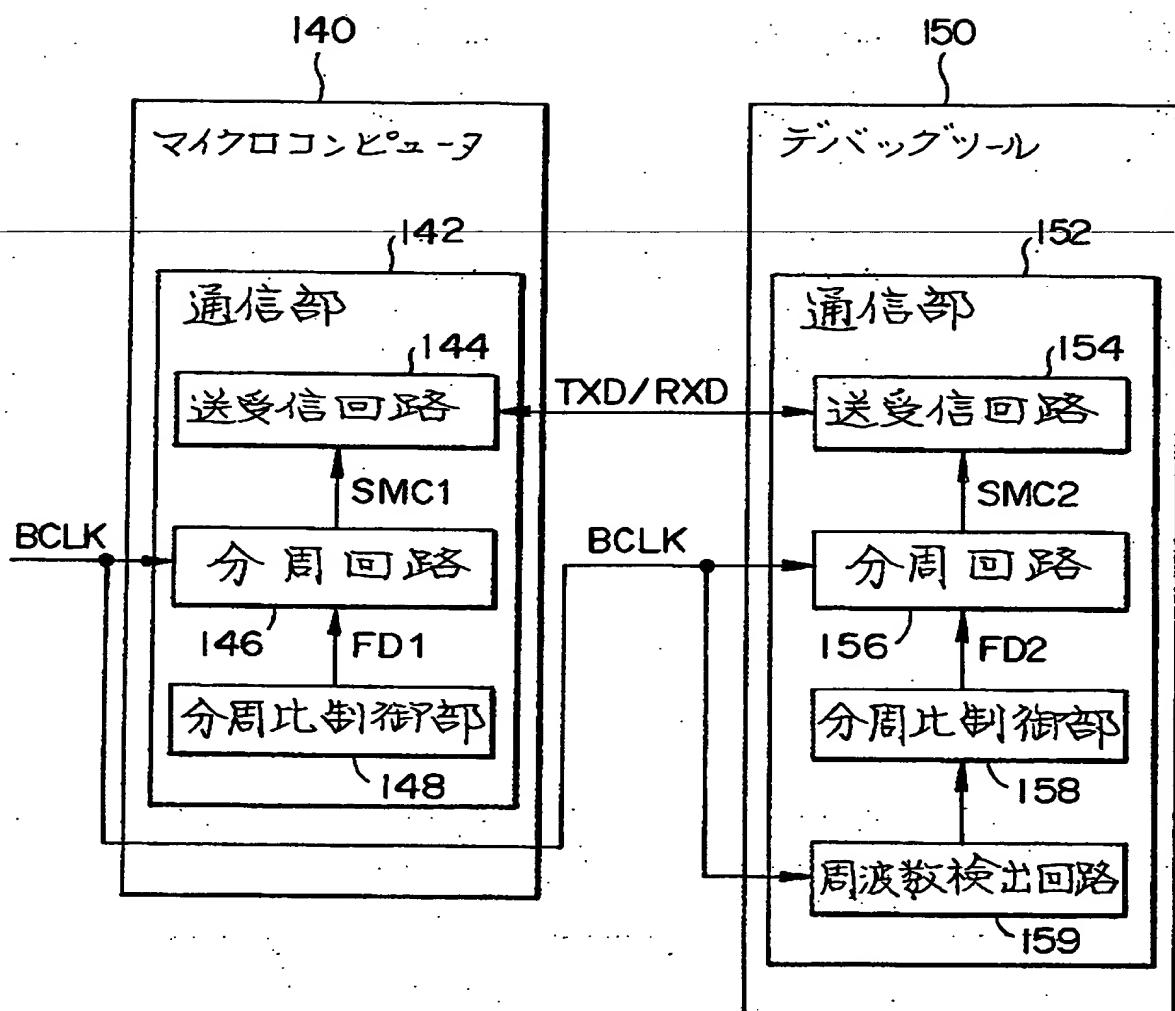
【図16】



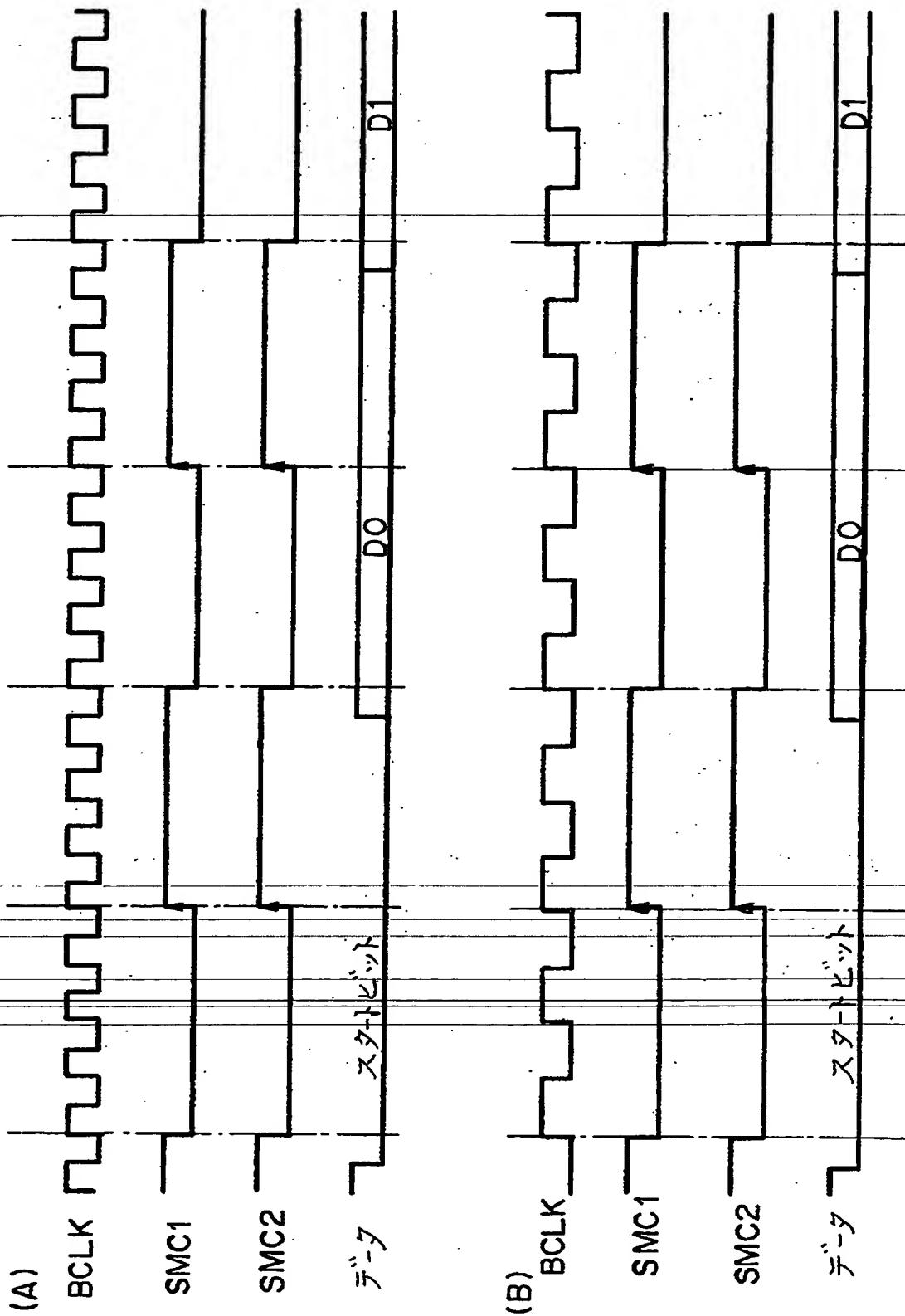
【図17】



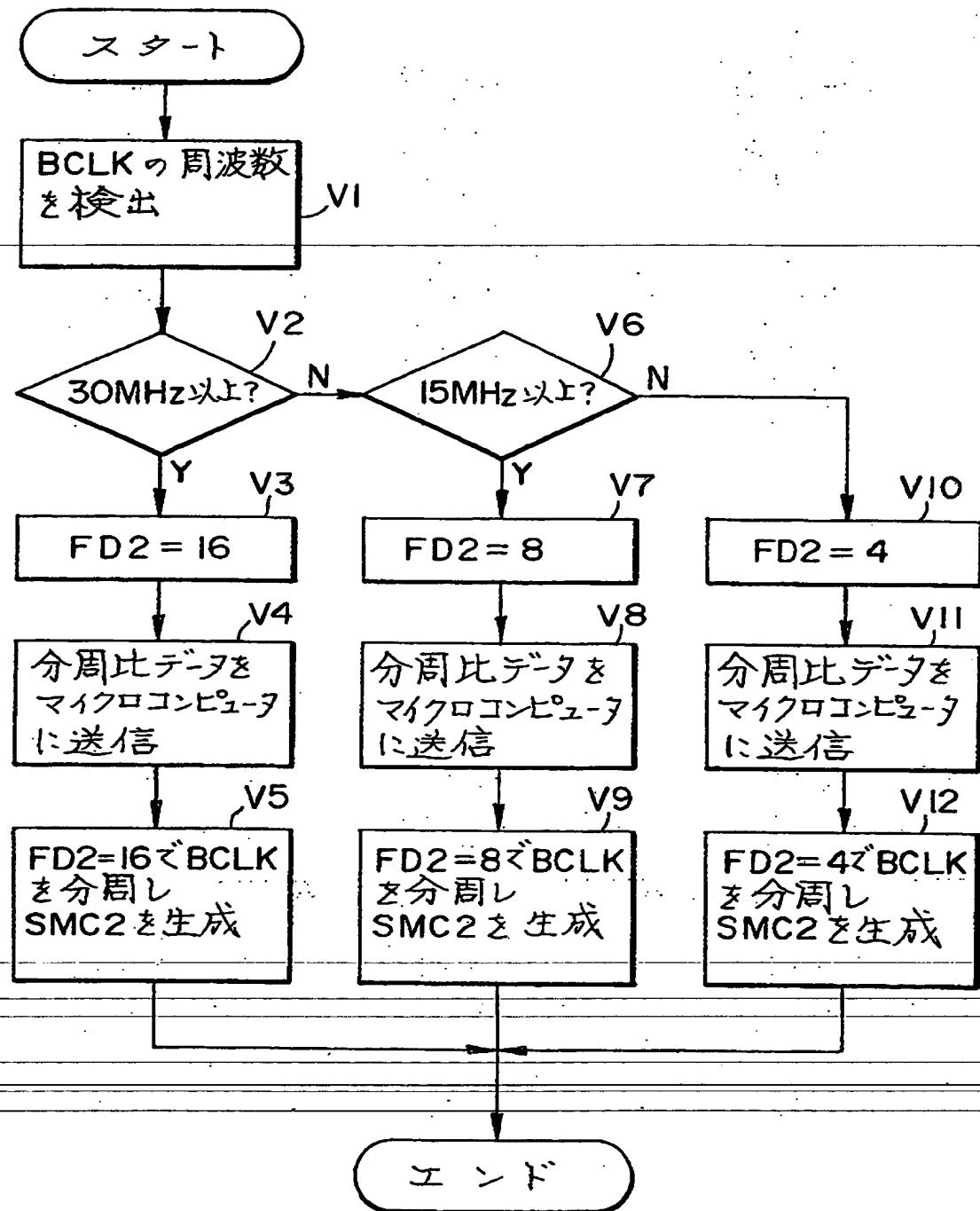
【図18】



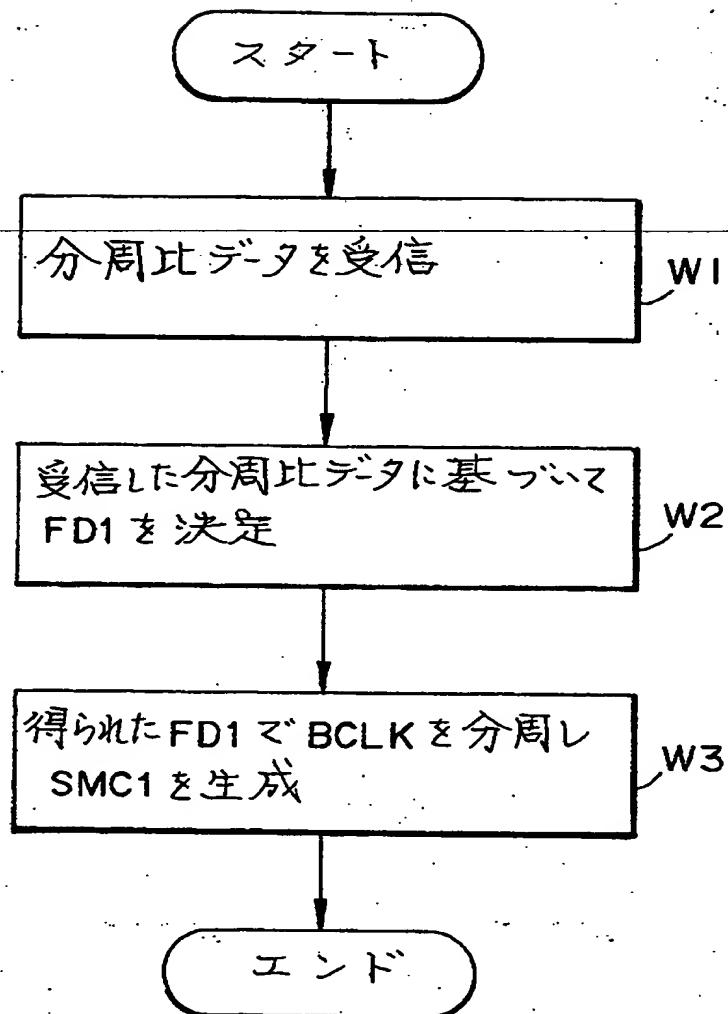
【図19】



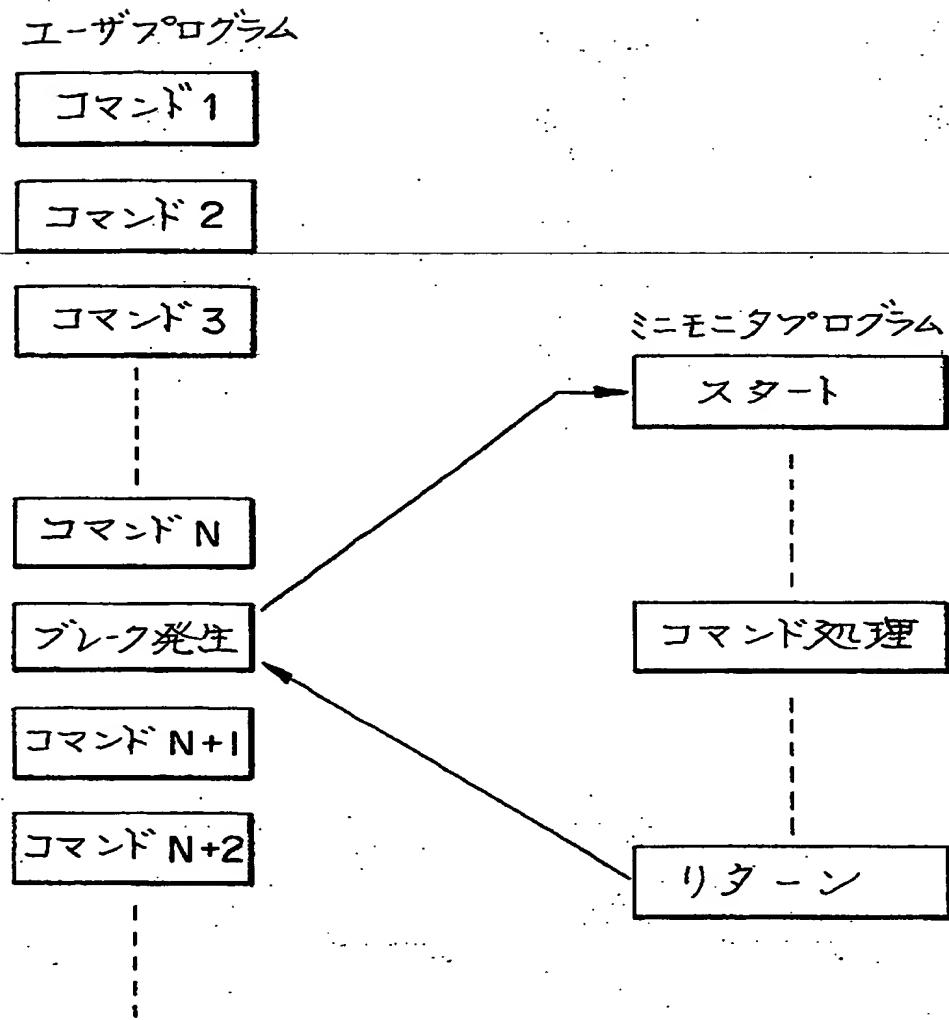
【図20】



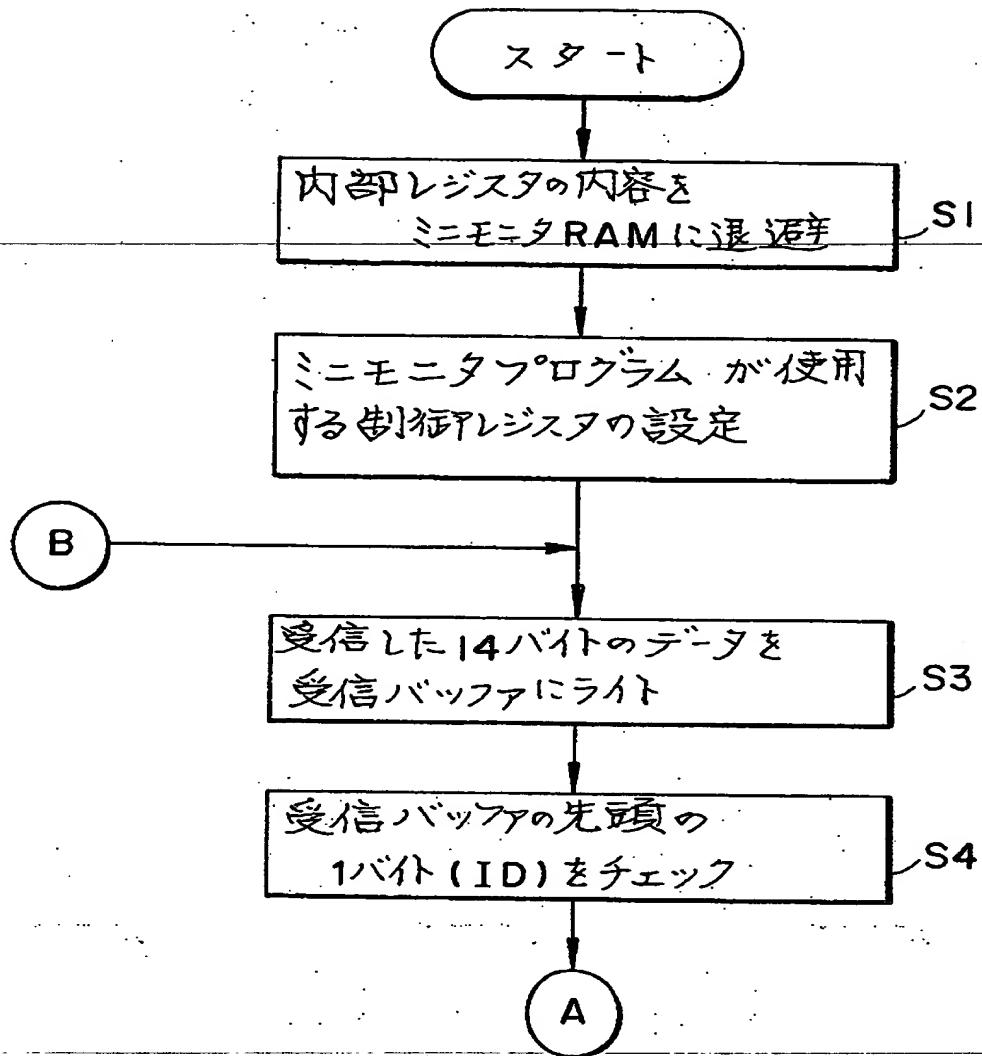
【図21】



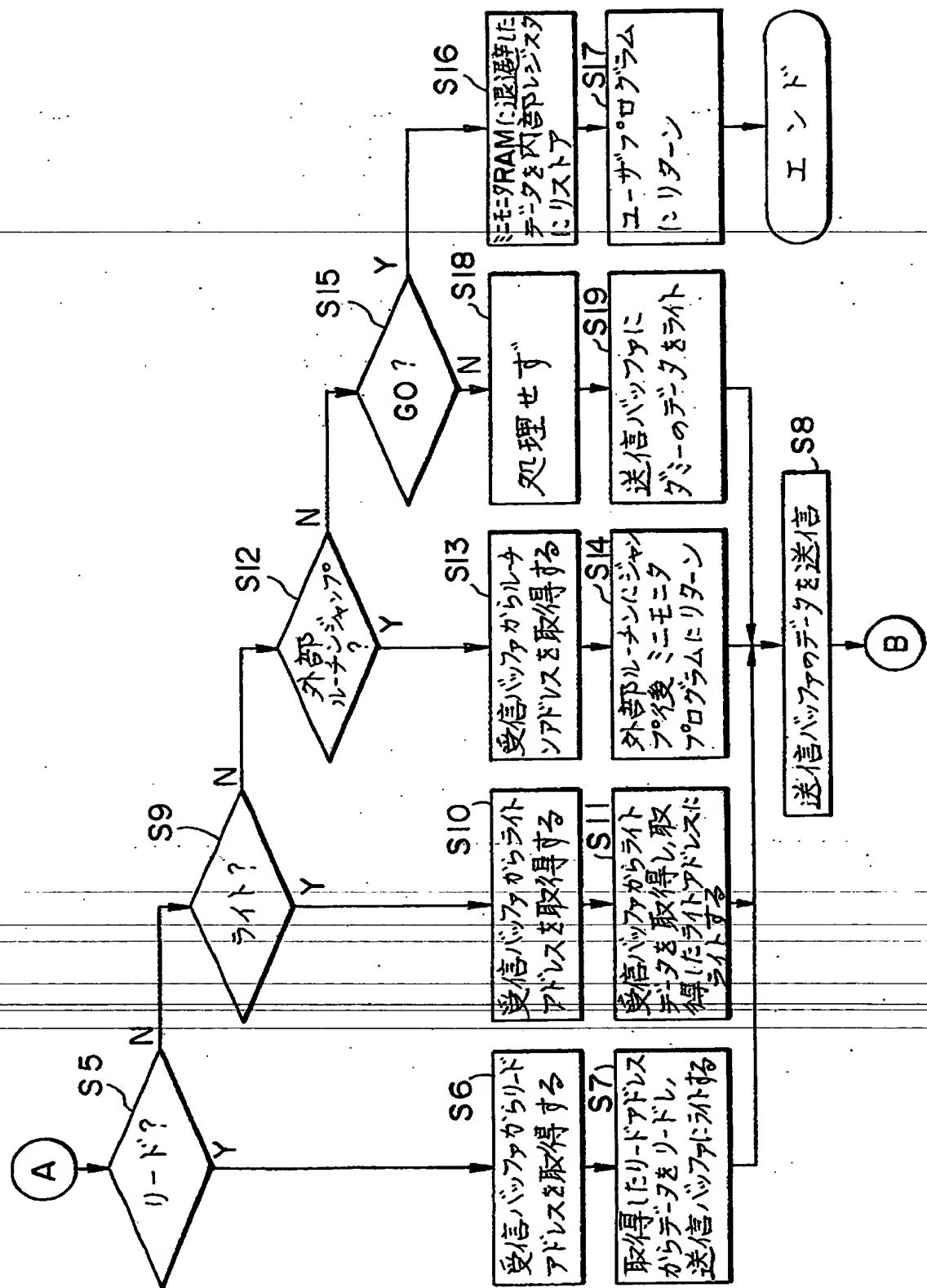
【図22】



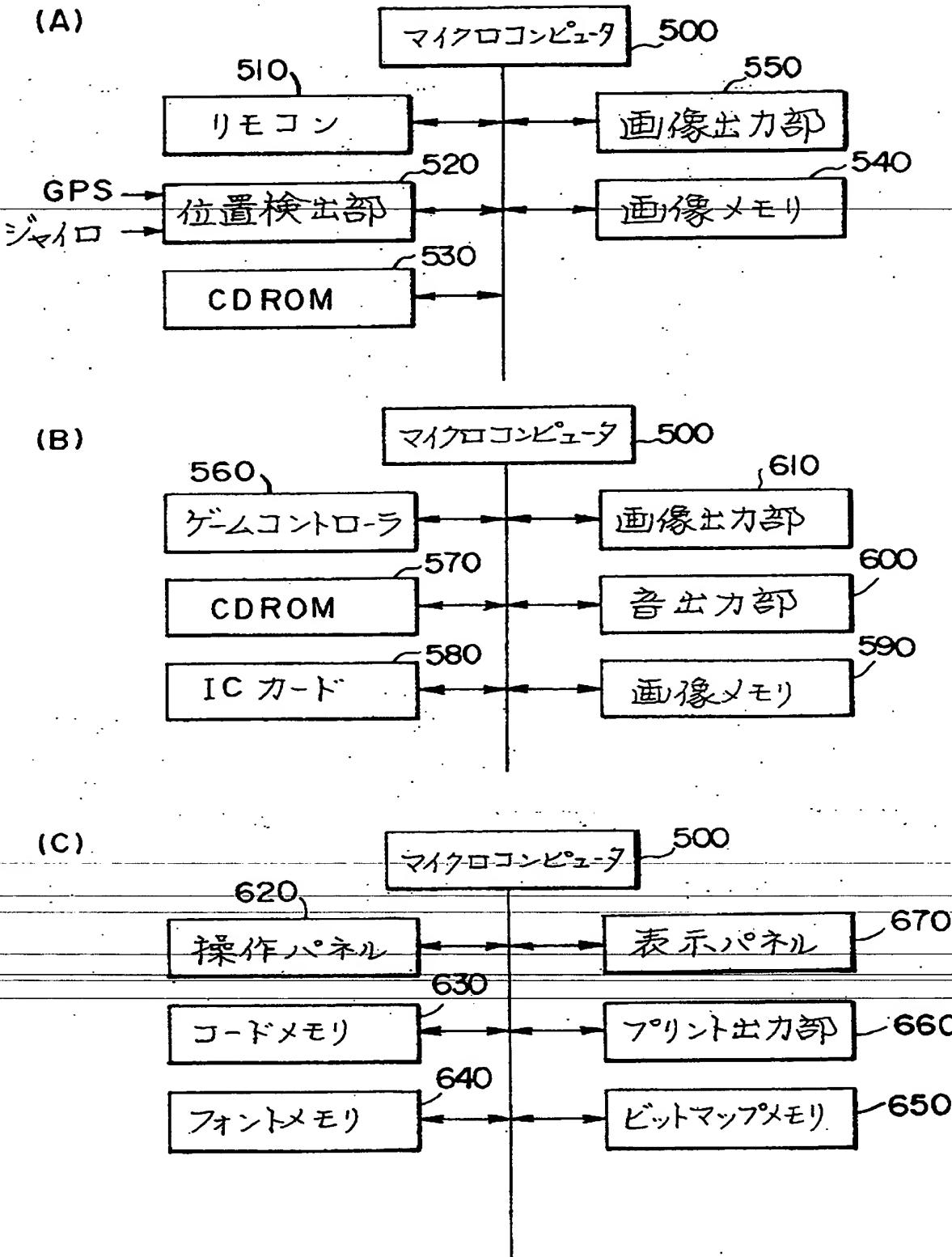
【図23】



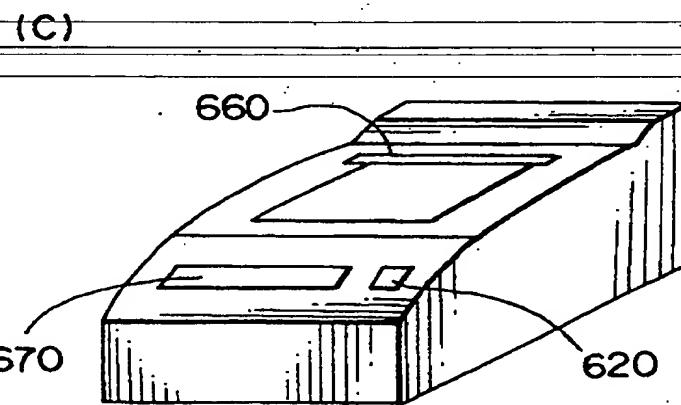
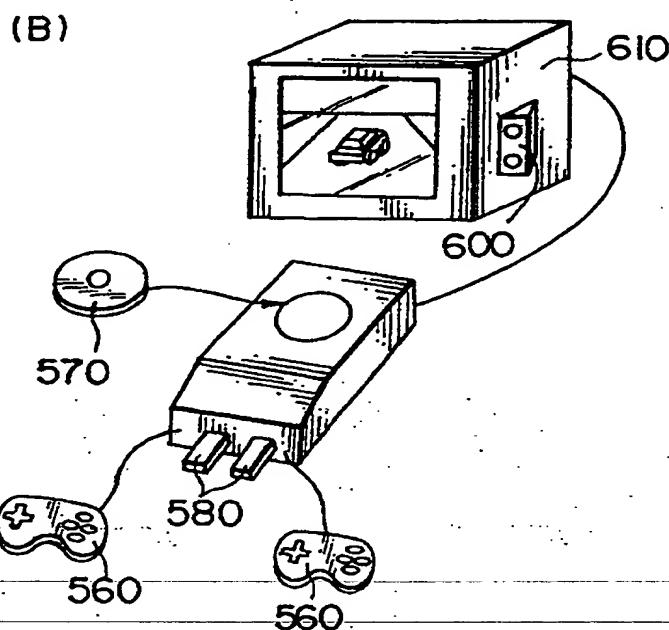
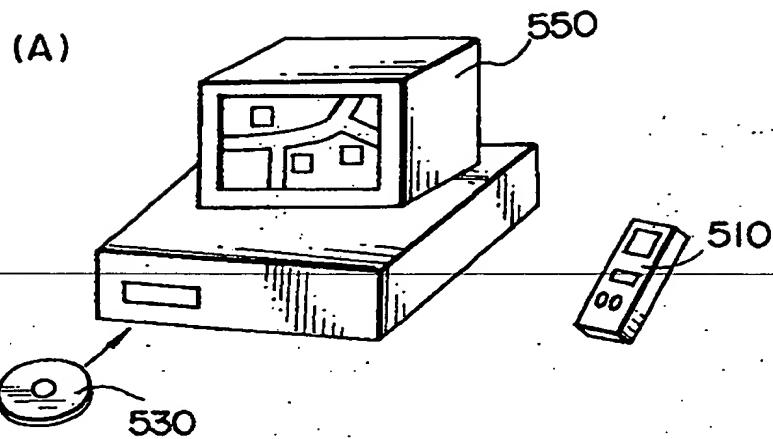
【図24】



【図25】



【図26】



【書類名】 要約書

【要約】

【課題】 小さな命令コードサイズ又は回路規模でオンチップデバッグ機能を実現できるマイクロコンピュータ、電子機器、デバッグシステムを提供すること。

【解決手段】 メインモニタ部16がデバッグコマンドをプリミティブコマンドに変換する。ミニモニタ部14はメインモニタ部16との間でデータを送受信し、受信データに基づき決定したプリミティブコマンドを実行する。プリミティブコマンドはGO、ライト、リードコマンドを含む。デバッグモード時のメモリマップ上にアドレスが割り付けられる制御レジスタ、ミニモニタRAMを設ける。スレーブとなるミニモニタ部14とマスタとなるメインモニタ部16を半2重の双方向通信ラインで接続し送受信データを固定長にする。受信データにコマンド識別データを含ませる。ミニモニタプログラムをROMに格納する。調歩同期式でデータの送受信をしながらサンプリングクロックを生成するためのクロックをミニモニタ部14とメインモニタ部16とで共有する。

【選択図】 図2

【書類名】 職權訂正データ
【訂正書類】 特許願

<認定情報・付加情報>

【特許出願人】

【識別番号】 000002369
【住所又は居所】 東京都新宿区西新宿2丁目4番1号
【氏名又は名称】 セイコーエプソン株式会社

【代理人】 申請人
【識別番号】 100090473
【住所又は居所】 東京都杉並区荻窪5丁目26番13号 荻窪TMビル2階 井上・布施合同特許事務所
【氏名又は名称】 井上 一

【代理人】 申請人
【識別番号】 100090387
【住所又は居所】 東京都杉並区荻窪5丁目26番13号 荻窪TMビル2階 井上・布施合同特許事務所
【氏名又は名称】 布施 行夫

【代理人】 申請人
【識別番号】 100090398
【住所又は居所】 東京都杉並区荻窪5丁目26番13号 荻窪TMビル2階 井上・布施合同特許事務所
【氏名又は名称】 大渕 美千栄

出願人履歴情報

識別番号 [000002369]

1. 変更年月日 1990年 8月20日

[変更理由] 新規登録

住 所 東京都新宿区西新宿2丁目4番1号

氏 名 セイコーエプソン株式会社